Impact of Non-optimal Checkpoint Intervals on Overall Application Efficiency in Cluster Computing: *A Simulation-Based Study*

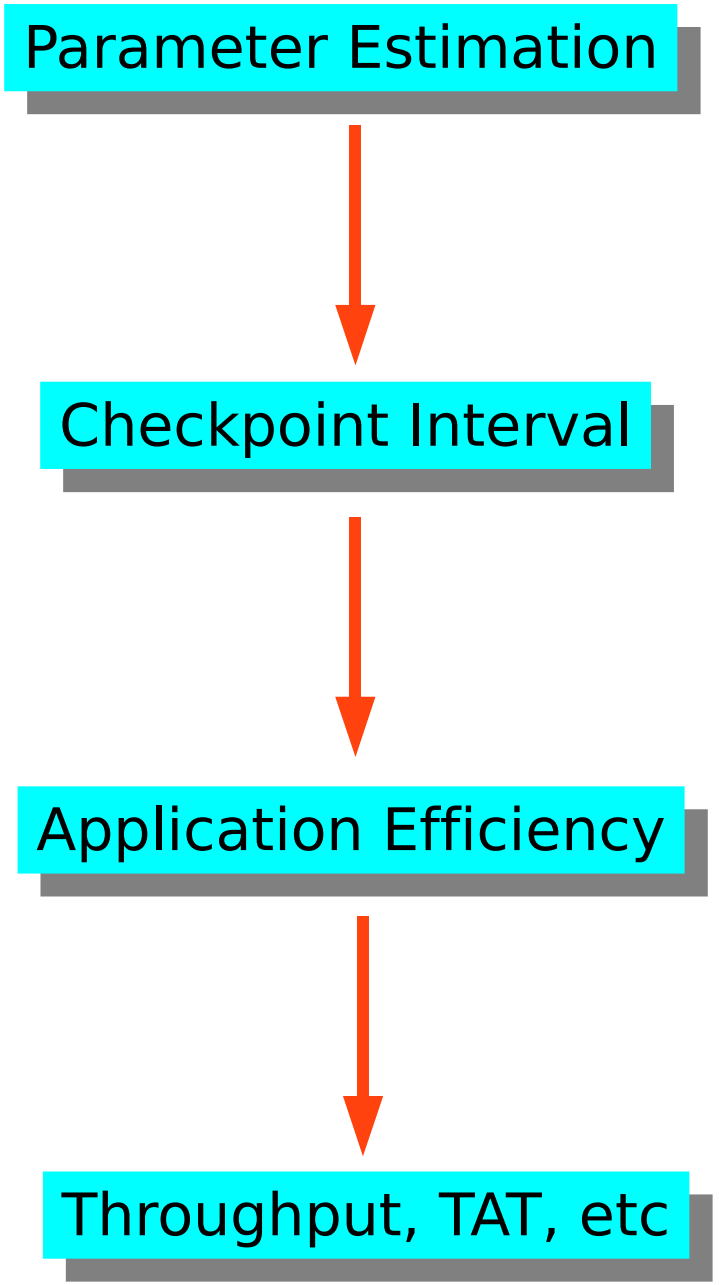
William M. Jones, PhD
Computer Science Department

Coastal Carolina University
Myrtle Beach, South Carolina

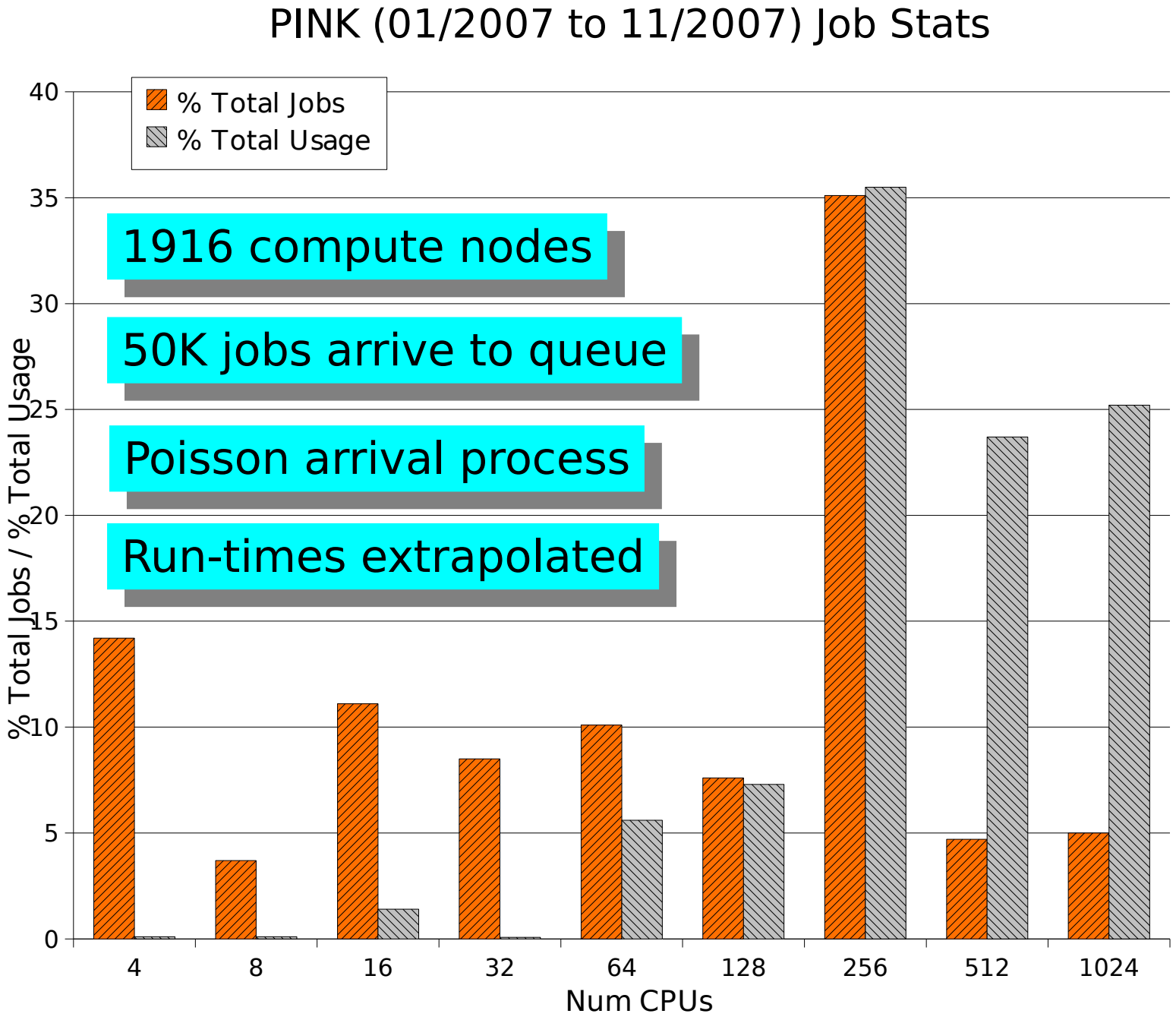
http://www.parl.clemson.edu/beosim

The Issue

- Application efficiency
  - Throughput
  - Turnaround times
  - Heavy loads
- Optimal checkpoint intervals
- AMTTI estimation
  - Running estimate
- Sensitivity to error
  - Non-optimal checkpoint interval
- Simulation-based study
  - Discrete event-driven
  - LANL's Pink Cluster

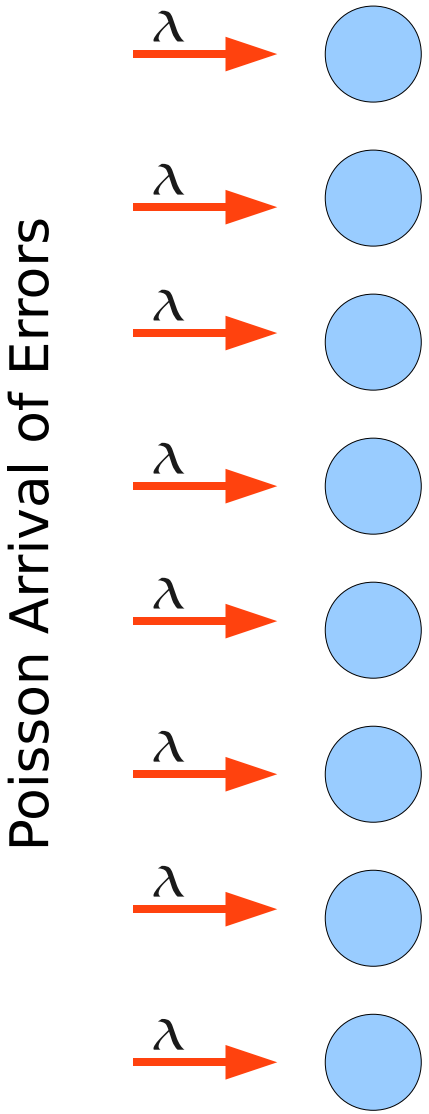


LANL's PINK Cluster Workload

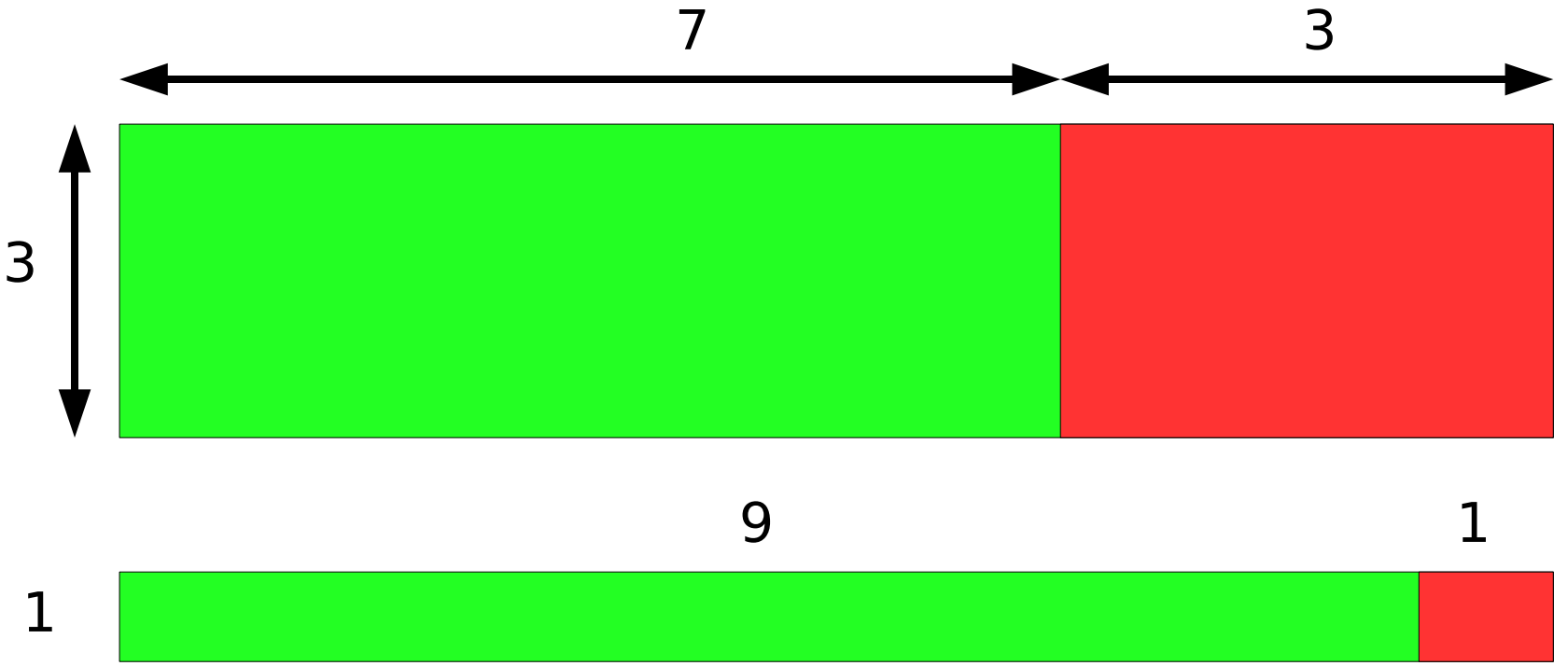


PINK Failure Model (Assumptions)

- Independent
- MTBF / node
- Single failure mode
- One failure
  - Halts at most 1 job
- Zero repair time
  - Job added to HOQ
- Checkpoint / Restart
  - 10 min dump time
  - 10 min restart time
- Simplistic
  - First pass



Checkpointing Metrics:
Average Application Efficiency



$$Avg.\ AE = \frac{\sum_{i=0}^{N-1} Ts_i n_i}{\sum_{i=0}^{N-1} Tr_i n_i} = .75$$

$$Avg.\ AE = \frac{\sum_{i=0}^{N-1} \frac{Ts_i}{Tr_i}}{N} = .8$$

Setting Checkpoint Interval For Each Job in Simulation

$$AMTTI_i = \frac{T}{n_i}, T = \frac{1}{\lambda}$$

Assume this

$$Tc_i = \sqrt{2\,\delta\,AMTTI_i}$$

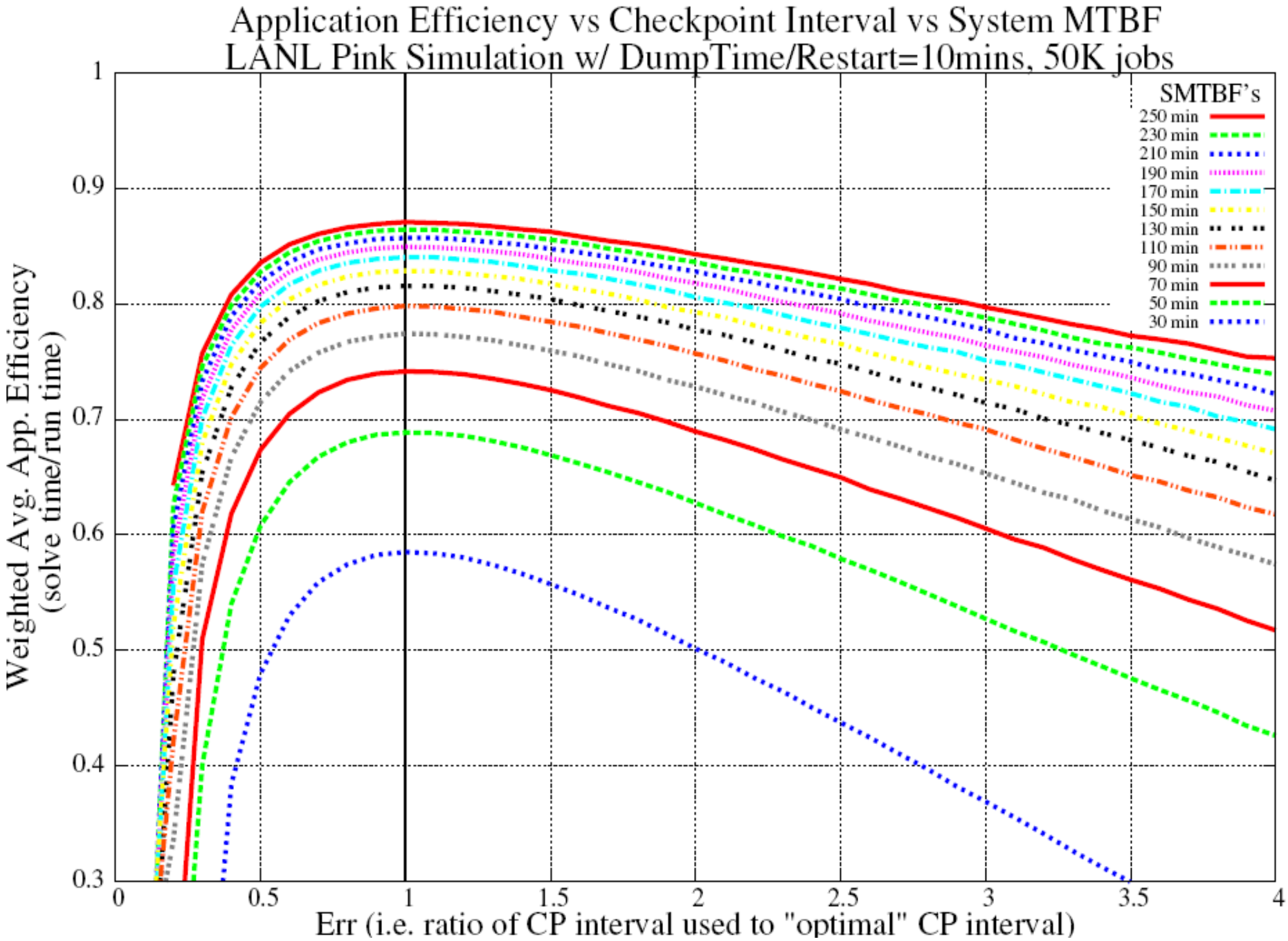
Approximate that

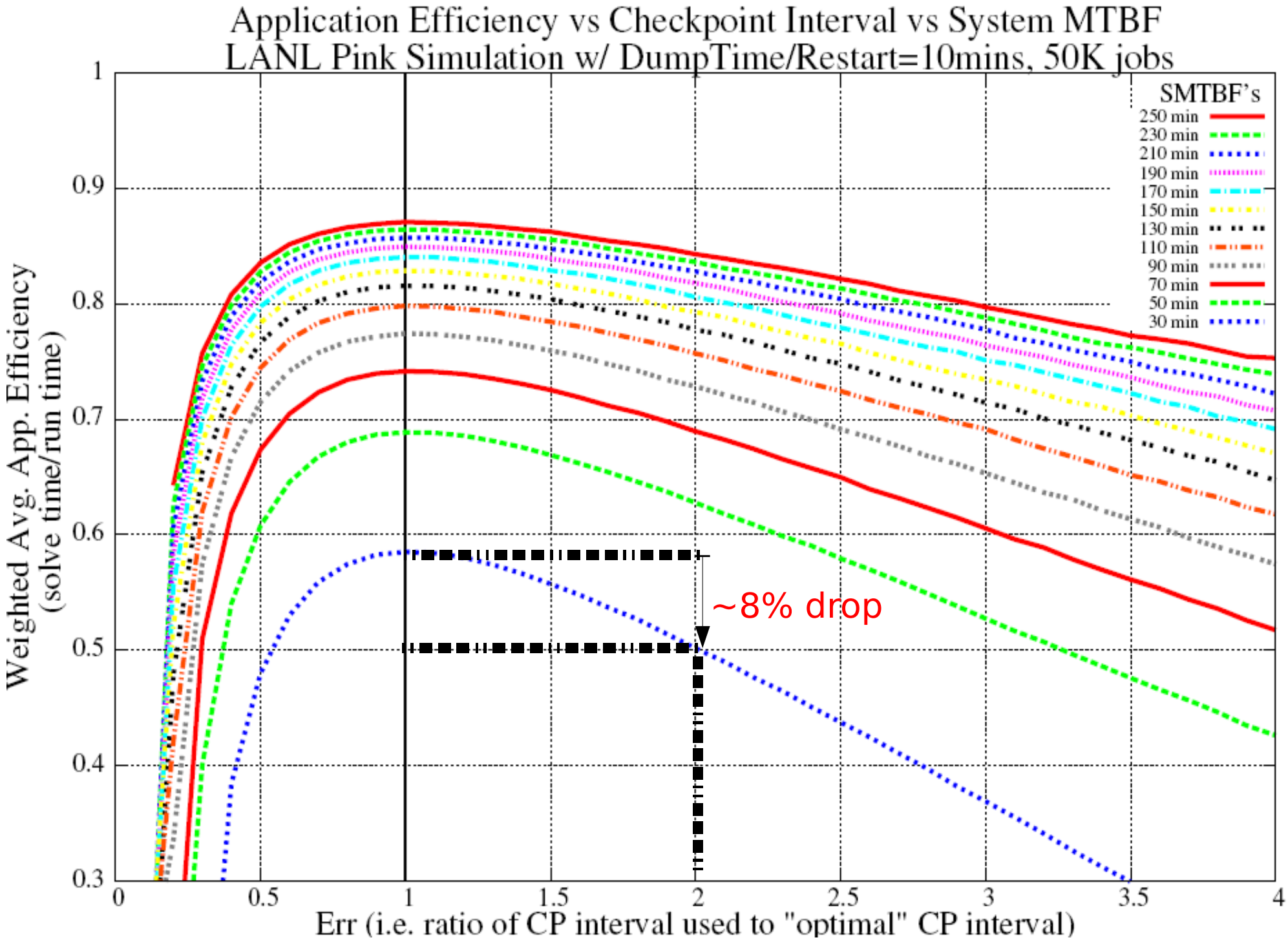
$$Tc_i = Tc_i * Err$$

Introduce error here

$$(Tc_i < \delta) \vee (Tc_i > Ts_i) \rightarrow no\,cp'ing$$

Is Tc logical for job?


Application Efficiency vs Checkpoint Interval vs System MTBF
LANL Pink Simulation w/ DumpTime/Restart=10mins, 50K jobs
Weighted Avg. App. Efficiency
(solve time/run time)
Err (i.e. ratio of CP interval used to "optimal" CP interval)
SMTBF's
250 min
230 min
210 min
190 min
170 min
150 min
130 min
110 min
90 min
70 min
50 min
30 min
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
0.5
1
1.5
2
2.5
3
3.5
4

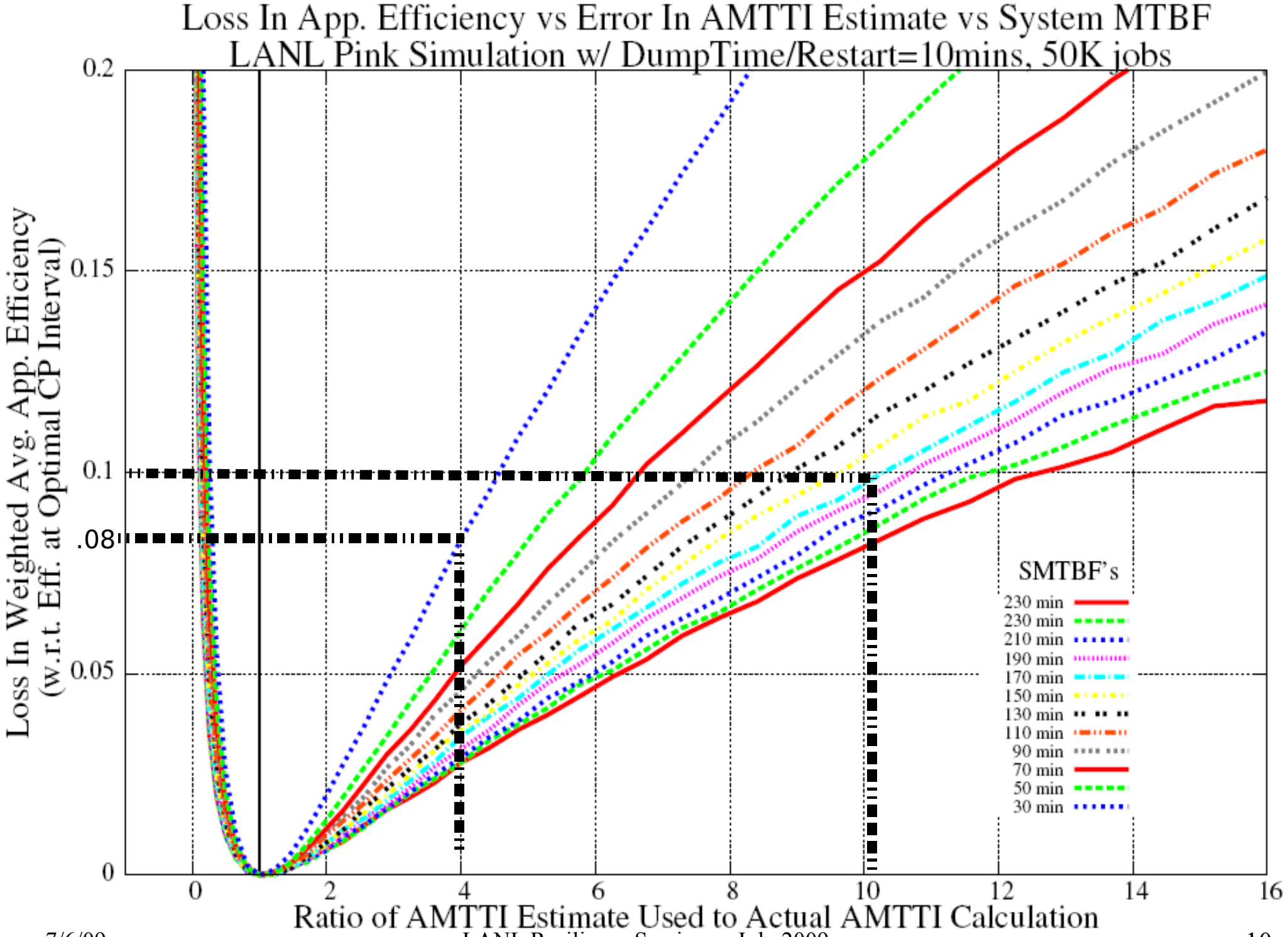

Application Efficiency vs Checkpoint Interval vs System MTBF
LANL Pink Simulation w/ DumpTime/Restart=10mins, 50K jobs
Weighted Avg. App. Efficiency
(solve time/run time)
Err (i.e. ratio of CP interval used to "optimal" CP interval)
SMTBF's
250 min
230 min
210 min
190 min
170 min
150 min
130 min
110 min
90 min
70 min
50 min
30 min
~8% drop
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
0.5
1
1.5
2
2.5
3
3.5
4

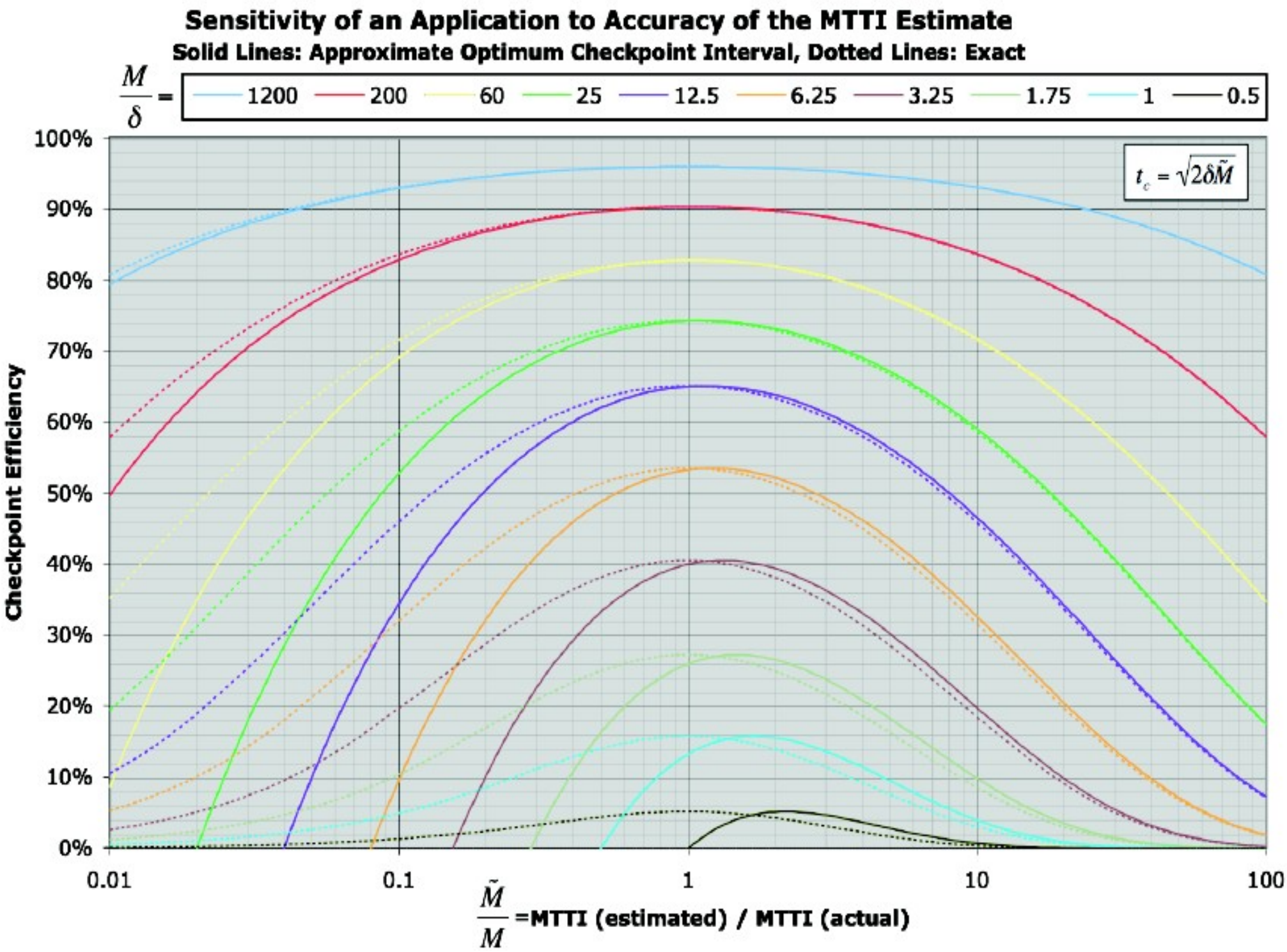
Back Calculate Loss of Application Efficiency as Function of Error in AMTTI

$$Tc_i = \sqrt{2\,\delta\,AMTTI_i}$$

Assume AMTTI is the sole source of error in Tc

A factor of 2 error in Tc would result from a factor of 4 error in AMTTI


Loss In App. Efficiency vs Error In AMTTI Estimate vs System MTBF
LANL Pink Simulation w/ DumpTime/Restart=10mins, 50K jobs
Loss In Weighted Avg. App. Efficiency
(w.r.t. Eff. at Optimal CP Interval)
0.2
0.15
0.1
.08
0.05
0
0
2
4
6
8
10
12
14
16
Ratio of AMTTI Estimate Used to Actual AMTTI Calculation
SMTBF's
230 min
230 min
210 min
190 min
170 min
150 min
130 min
110 min
90 min
70 min
50 min
30 min


Sensitivity of an Application to Accuracy of the MTTI Estimate
Solid Lines: Approximate Optimum Checkpoint Interval, Dotted Lines: Exact
M/δ =
1200
200
60
25
12.5
6.25
3.25
1.75
1
0.5
$t_c = \sqrt{2\delta\tilde{M}}$
Checkpoint Efficiency
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0.01
0.1
1
10
100
$\frac{\tilde{M}}{M}$ =MTTI (estimated) / MTTI (actual)

Daly, J. T., "Methodology and Metrics for Quantifying Application Throughput", Proceedings of NECDC '06

Daly states "... we do not need to be overly concerned about the precision of our checkpoint restart interval approximation [...] or our measurement of its dependent dump time and application MTTI parameters."

Application Efficiency Still Matters

Heavy System Load

Dramatic Impact on Queue Time

Turnaround Time = Queue Time + Execution Time

Besides

Energy Usage

But what about at larger scales where dumptime >> AMTTI ?

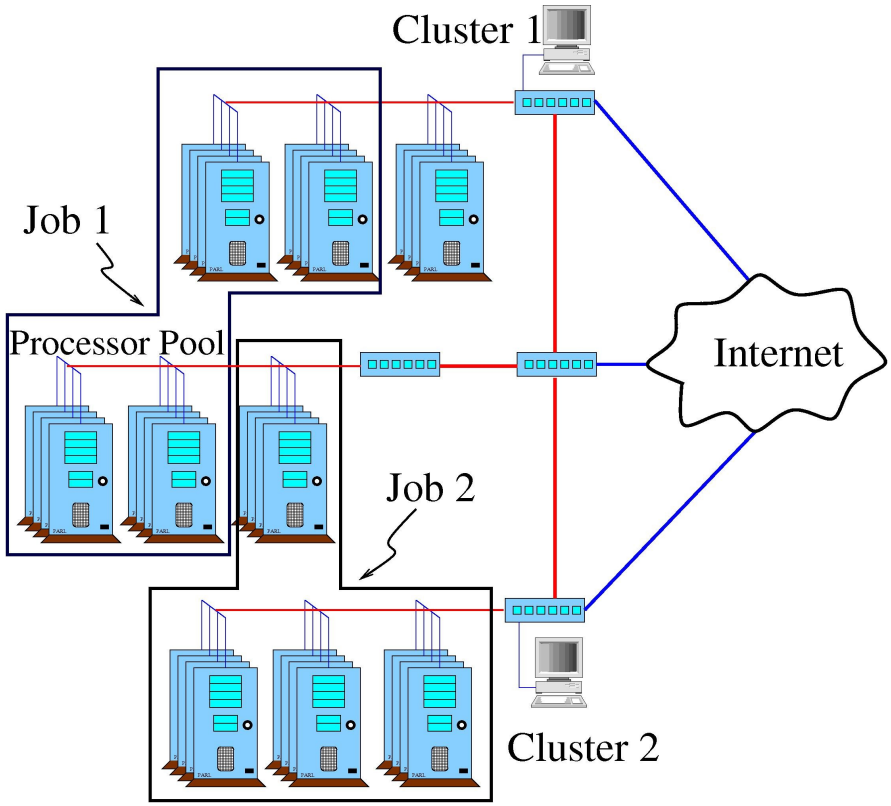
Future Work

- New cluster (larger systems) runs
  - Realistic failure stats
- Add new failure modes
  - Failures that impact more than one job
  - AMTTI << SMTBF
- Add AMTTI on-line parameter estimator
  - Initialize to a guess
  - As AMTTI is refined, Tc becomes closer to optimal
- Map jobs with resilience in mind
  - Schedule contiguous across switch, etc

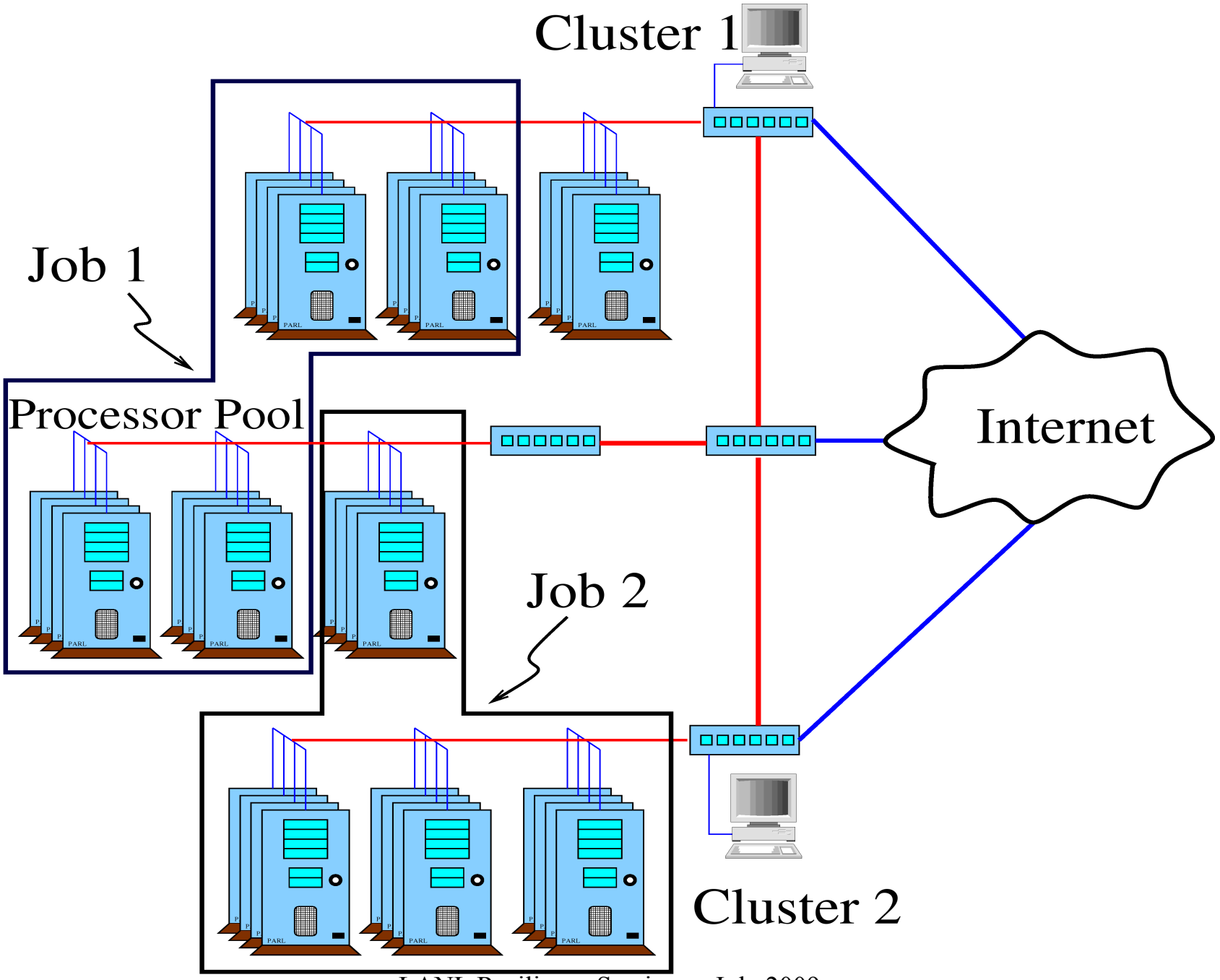
PART 2 of Presentation
Checkpoint / Migration Job Scheduling Next

Parallel Job Scheduling

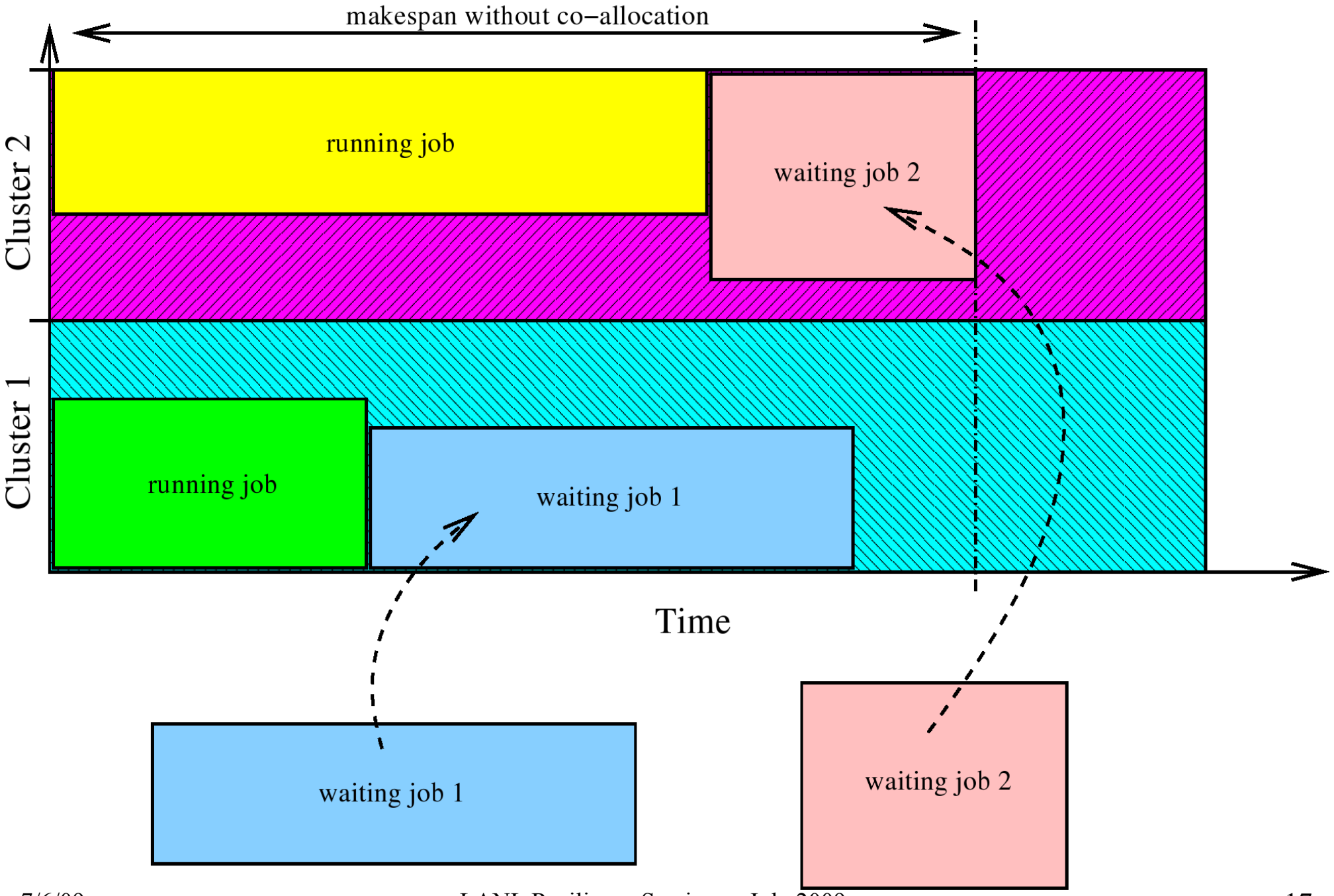
- Local execution
- Job migration
- **Job co-allocation**
  - Map across boundaries
  - Sharing resources
  - Network BW contention
- Can help or hurt
- Some example scenarios ...



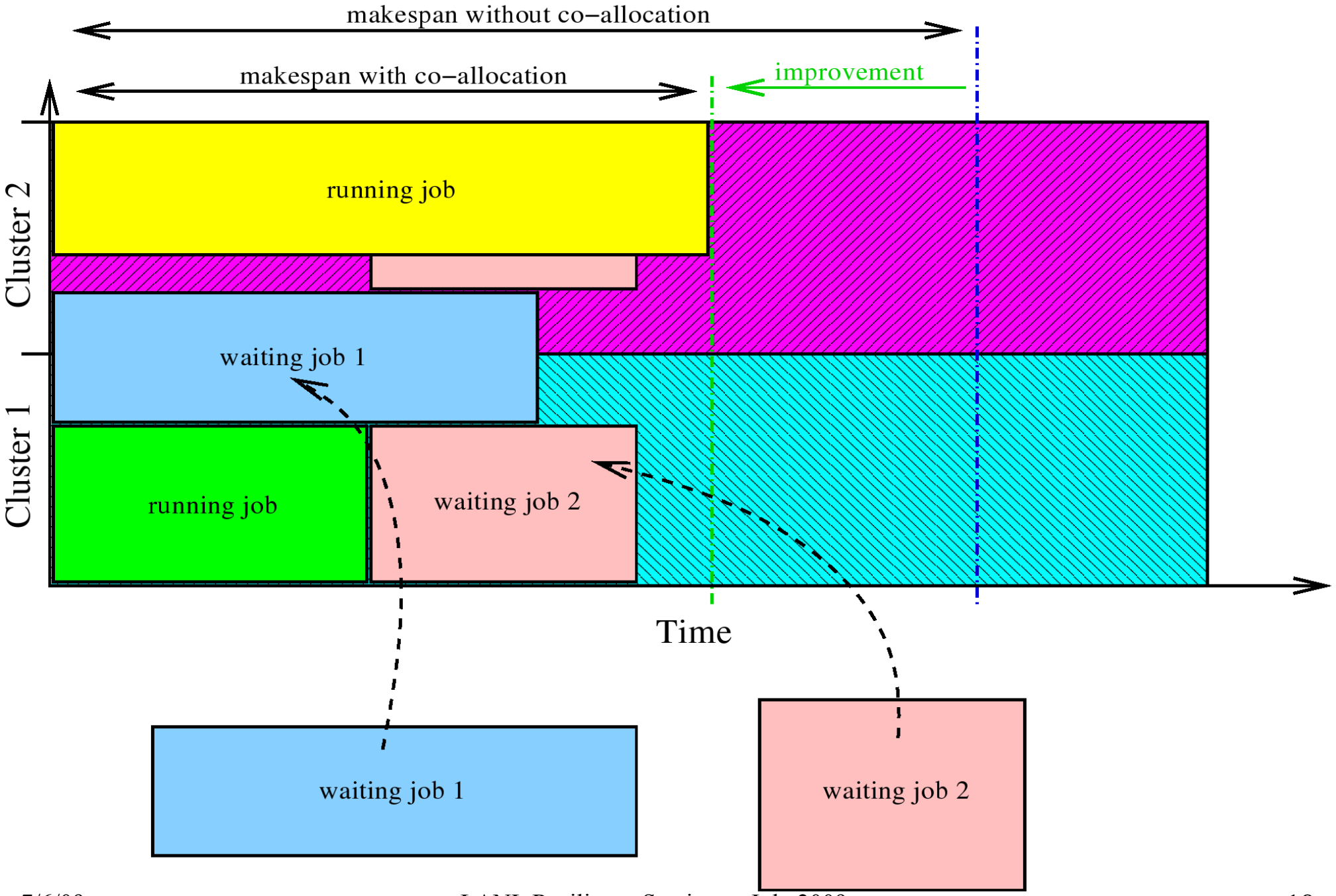
Multi-site Co-allocation



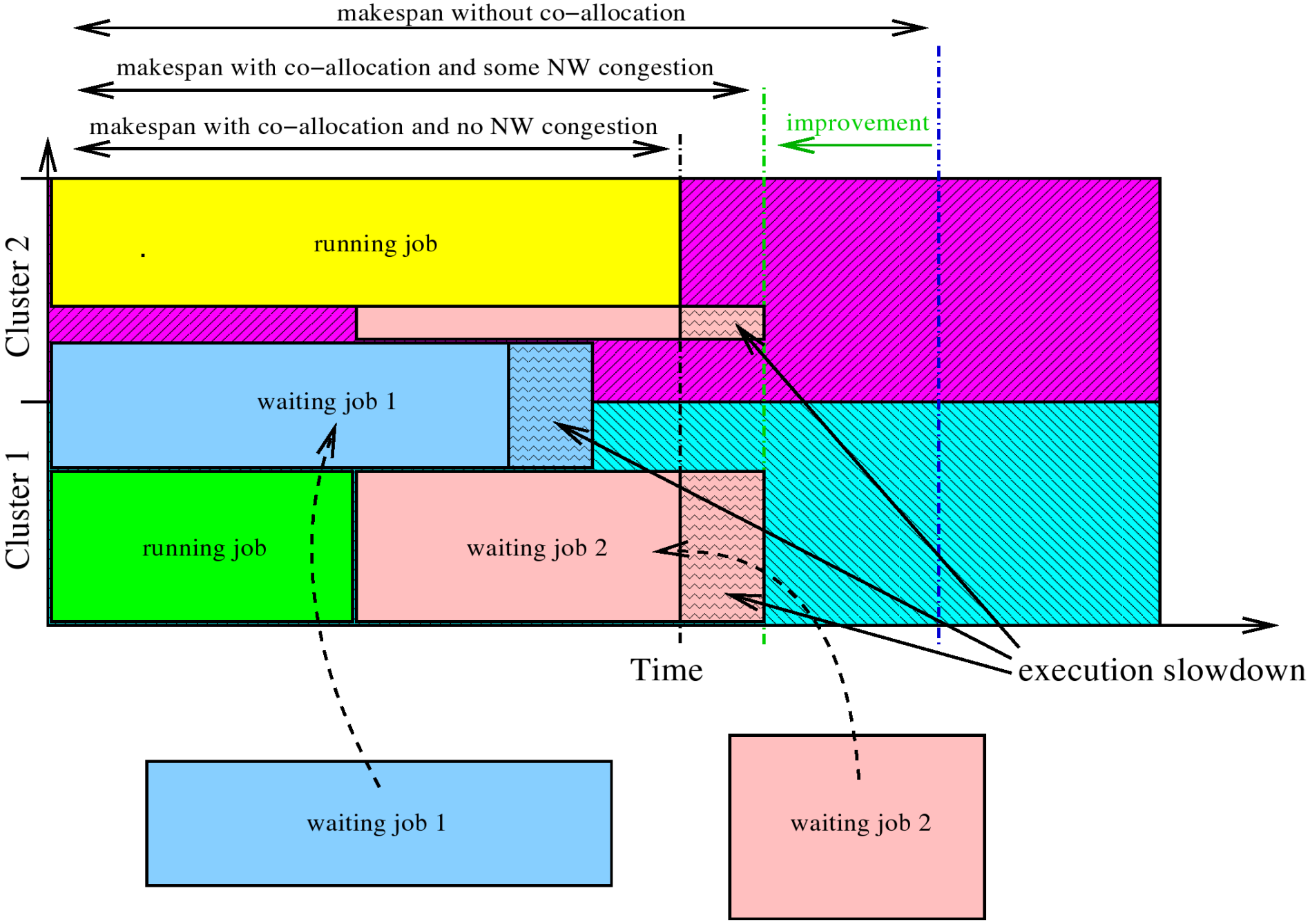
Scheduling w/o co-allocation



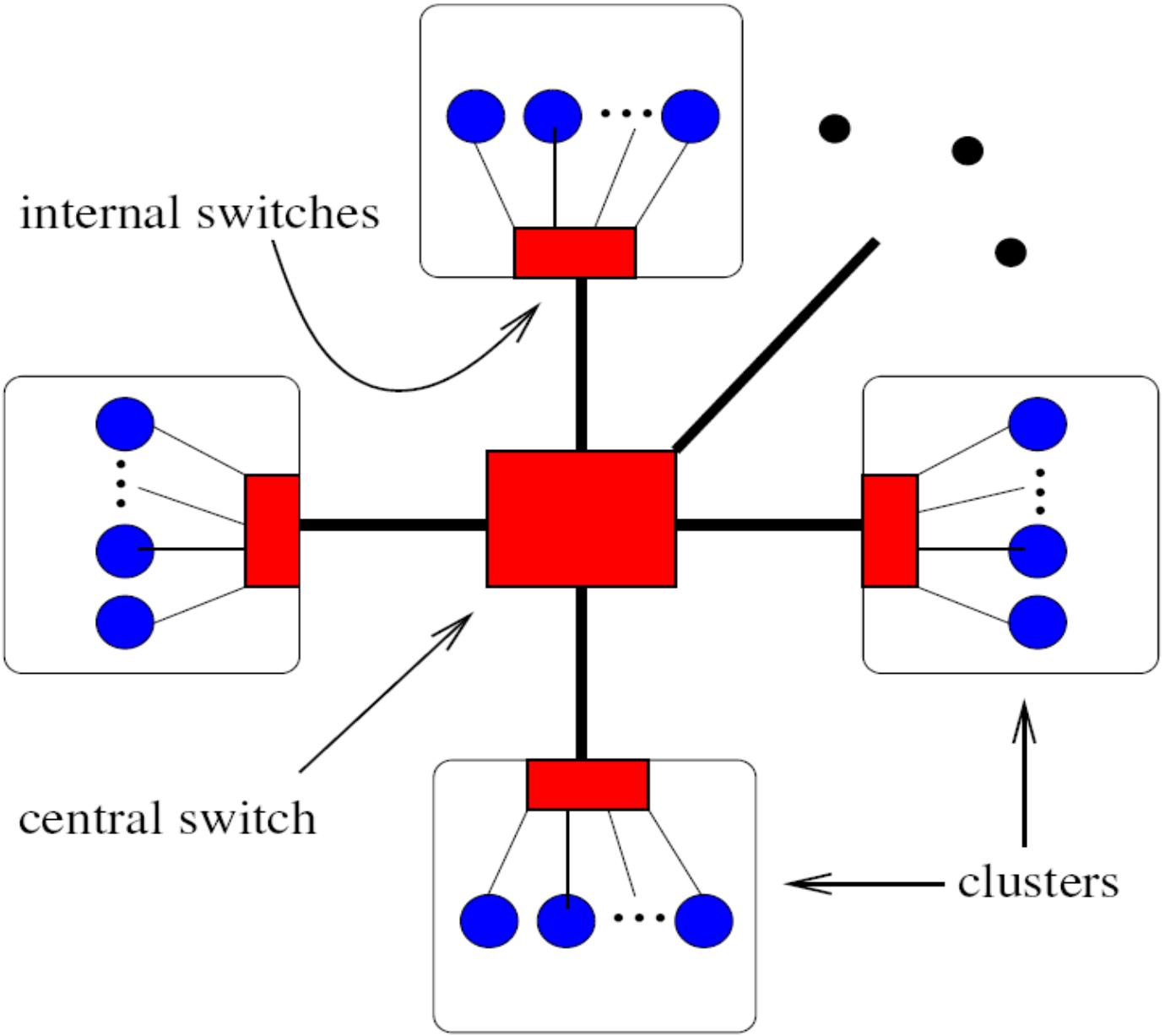
Scheduling w/ co-allocation



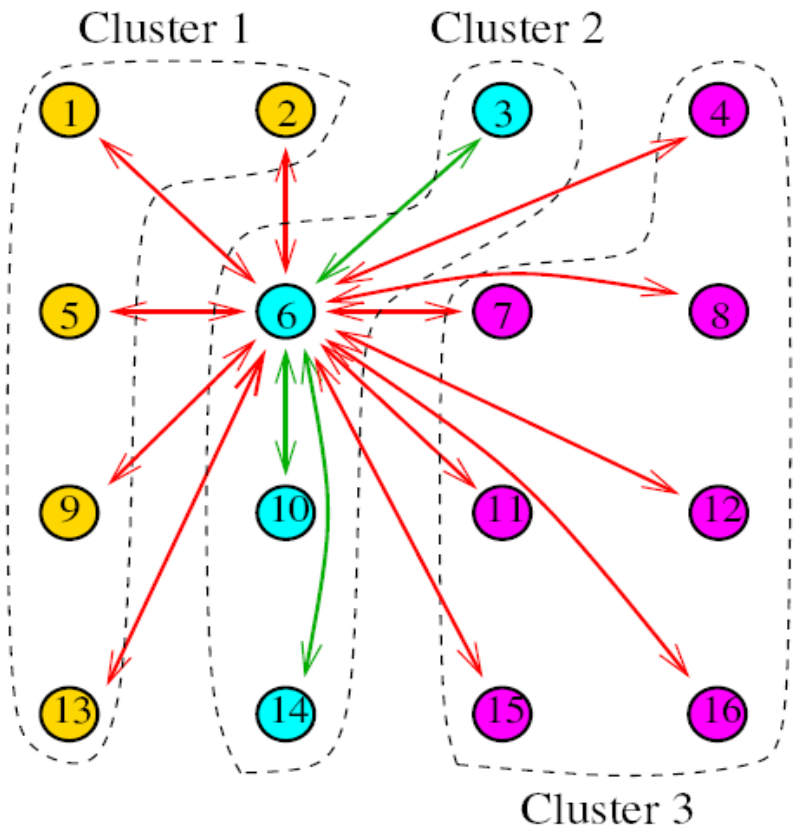
Co-allocation w/ slowdown



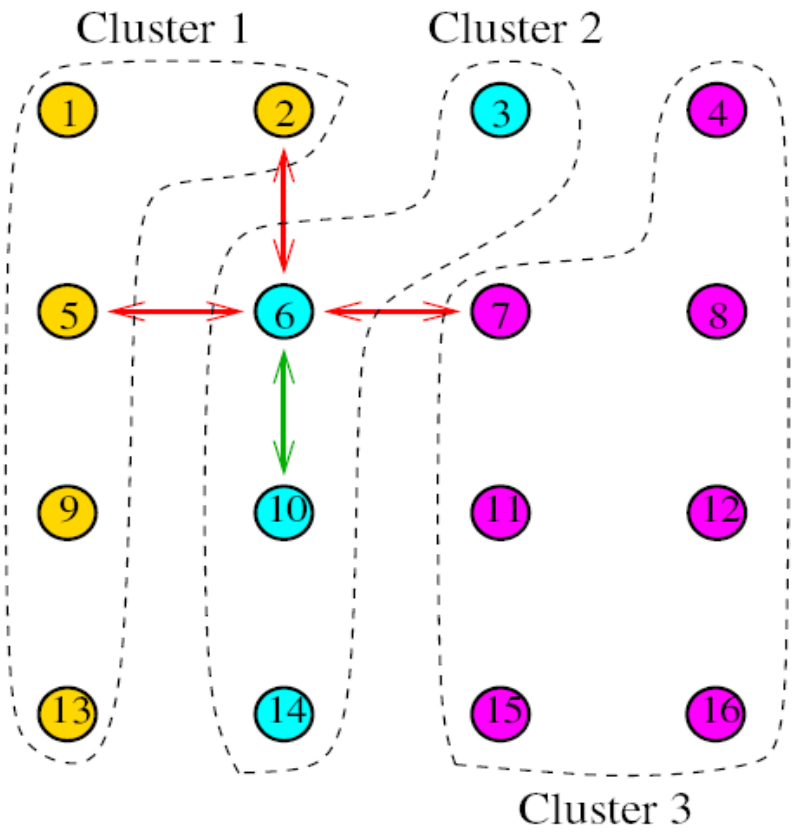
Idealized Multi-cluster Model



IPC Pattern Model

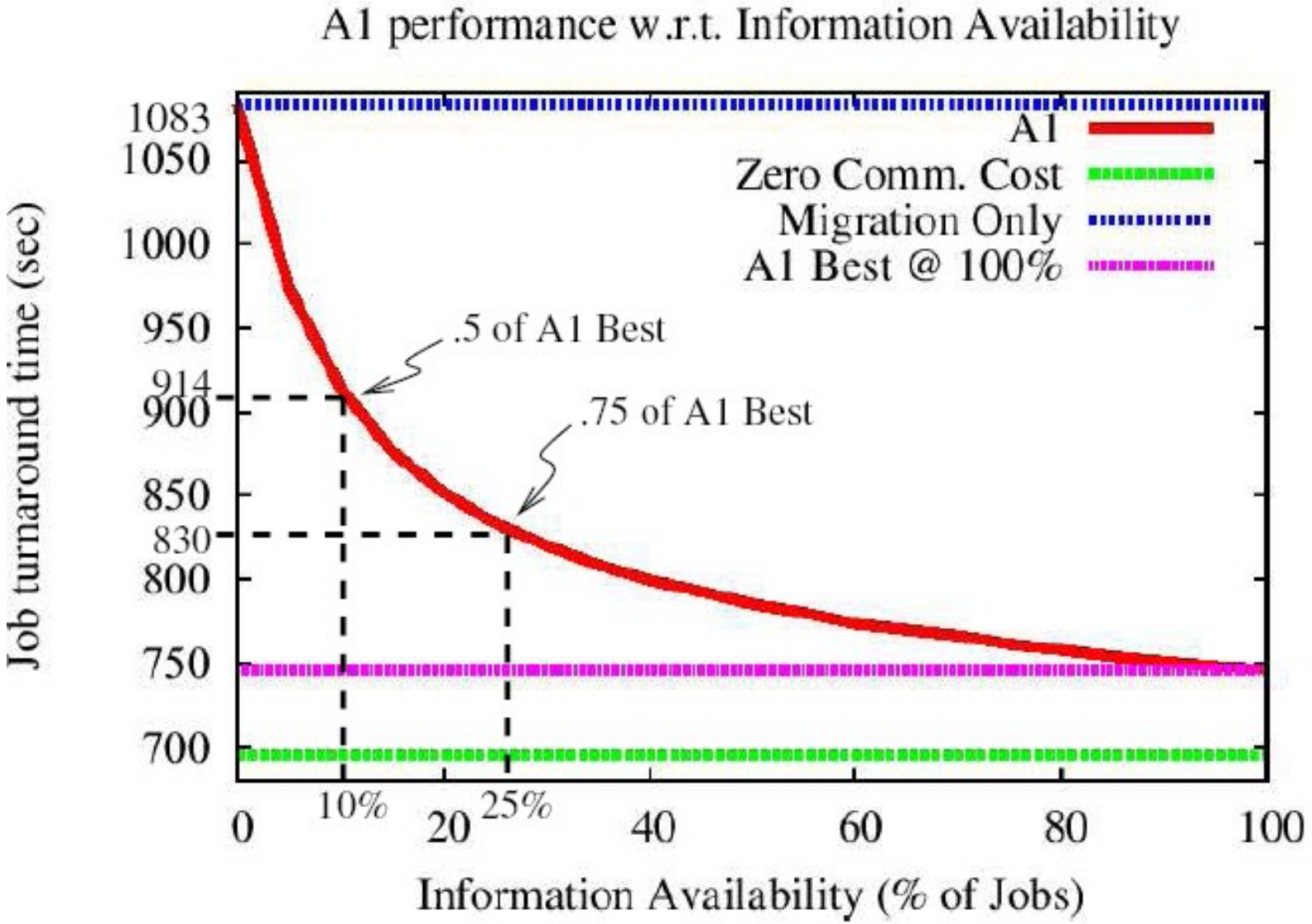


All-to-all personalized



2D

How Much Info Is "Enough"

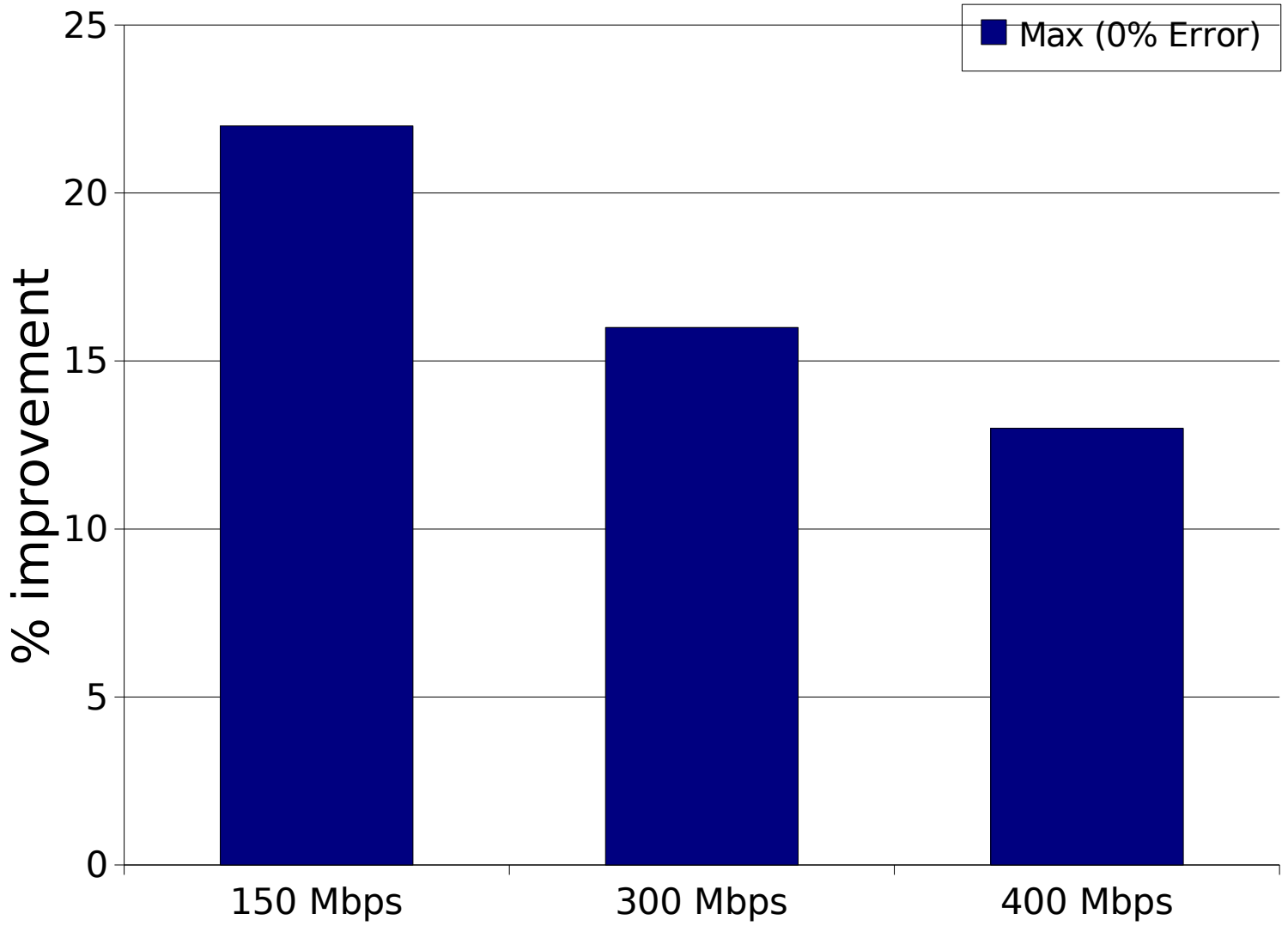


Max improvement possible: 36%

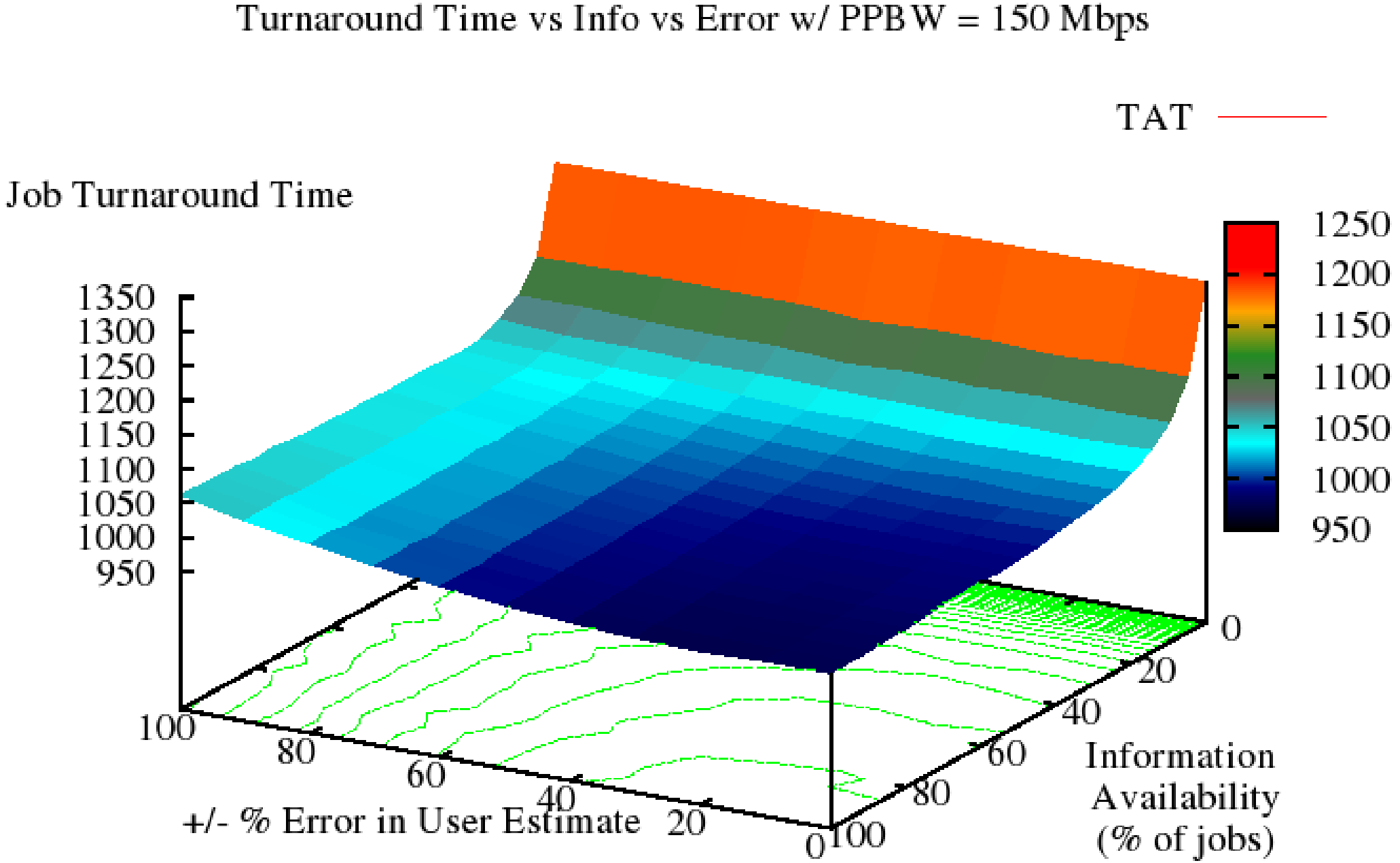
A1 Best: 31%

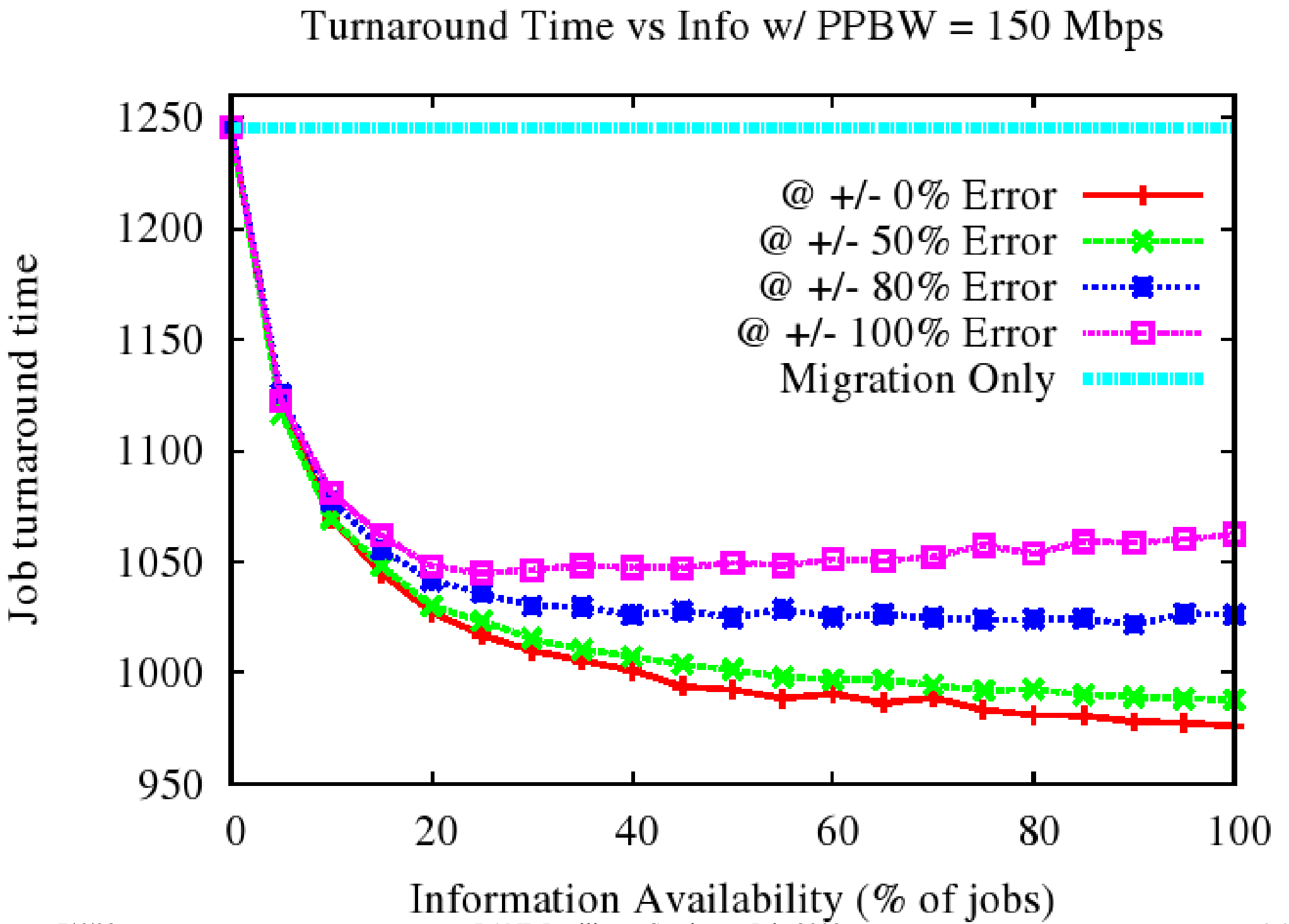
Summary of Previous Results

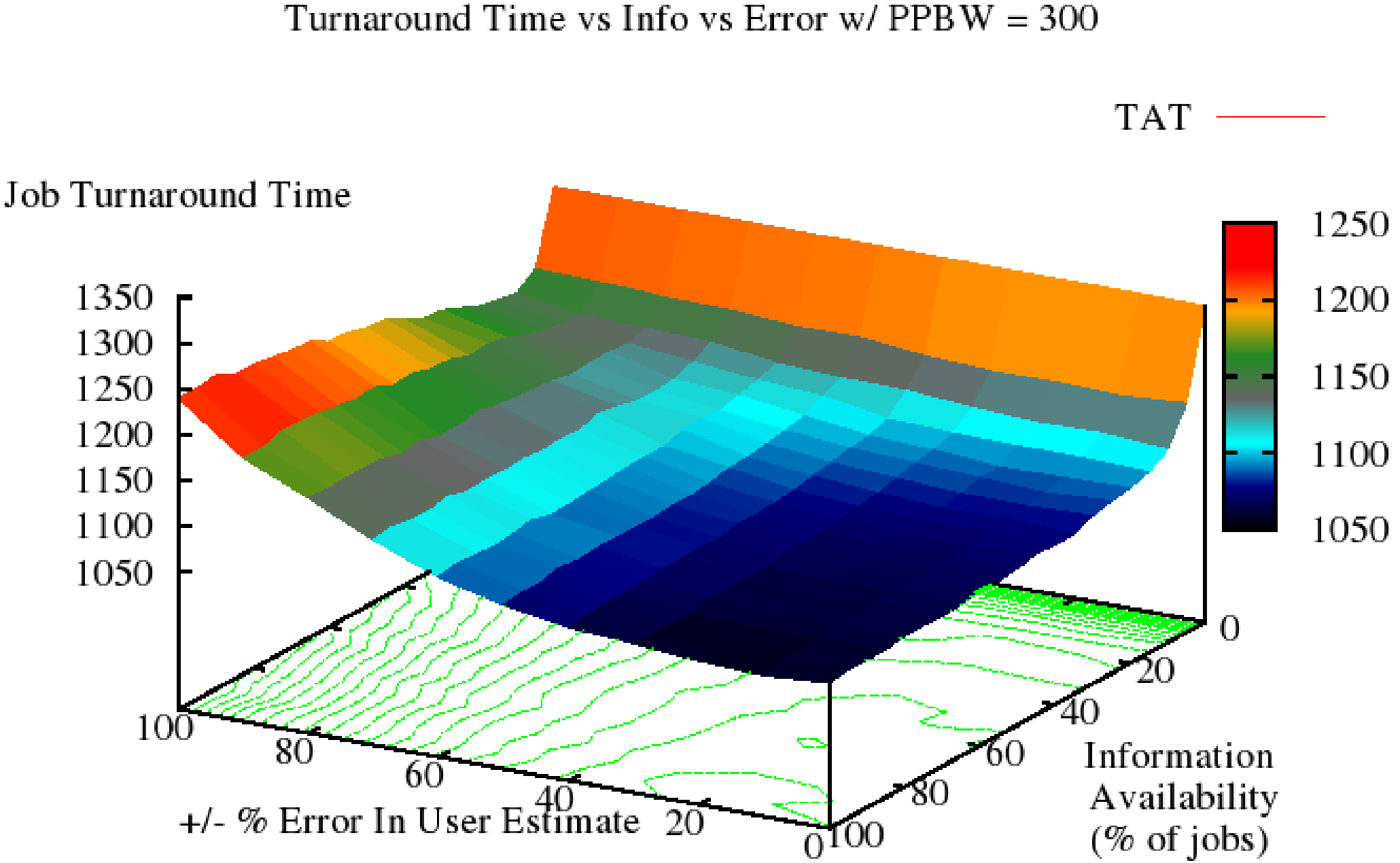
Improvement Over Migration Only w/ No Estimate Inaccuracy

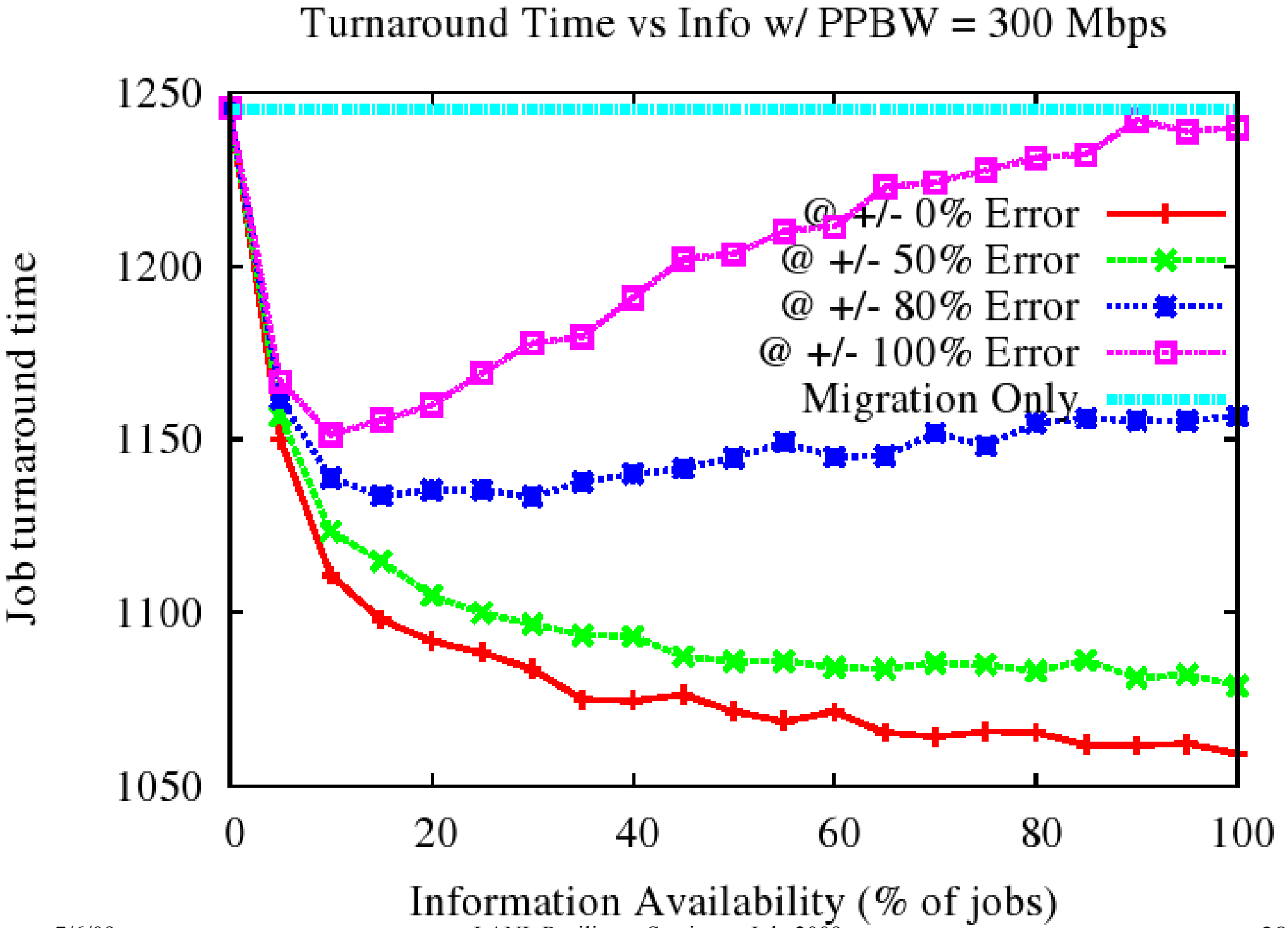


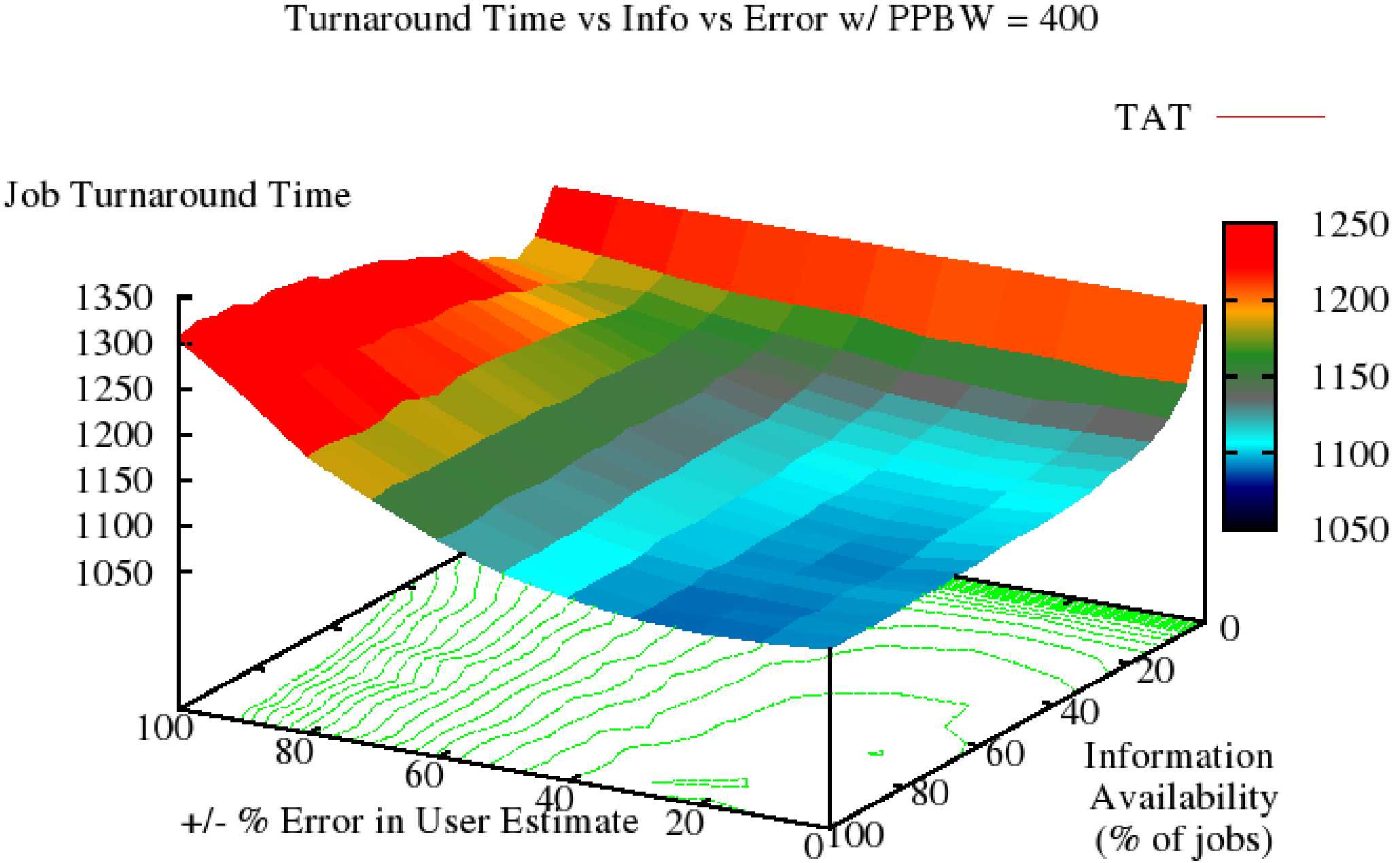
What happens if there is significant inaccuracy in the user-provided bandwidth estimates?

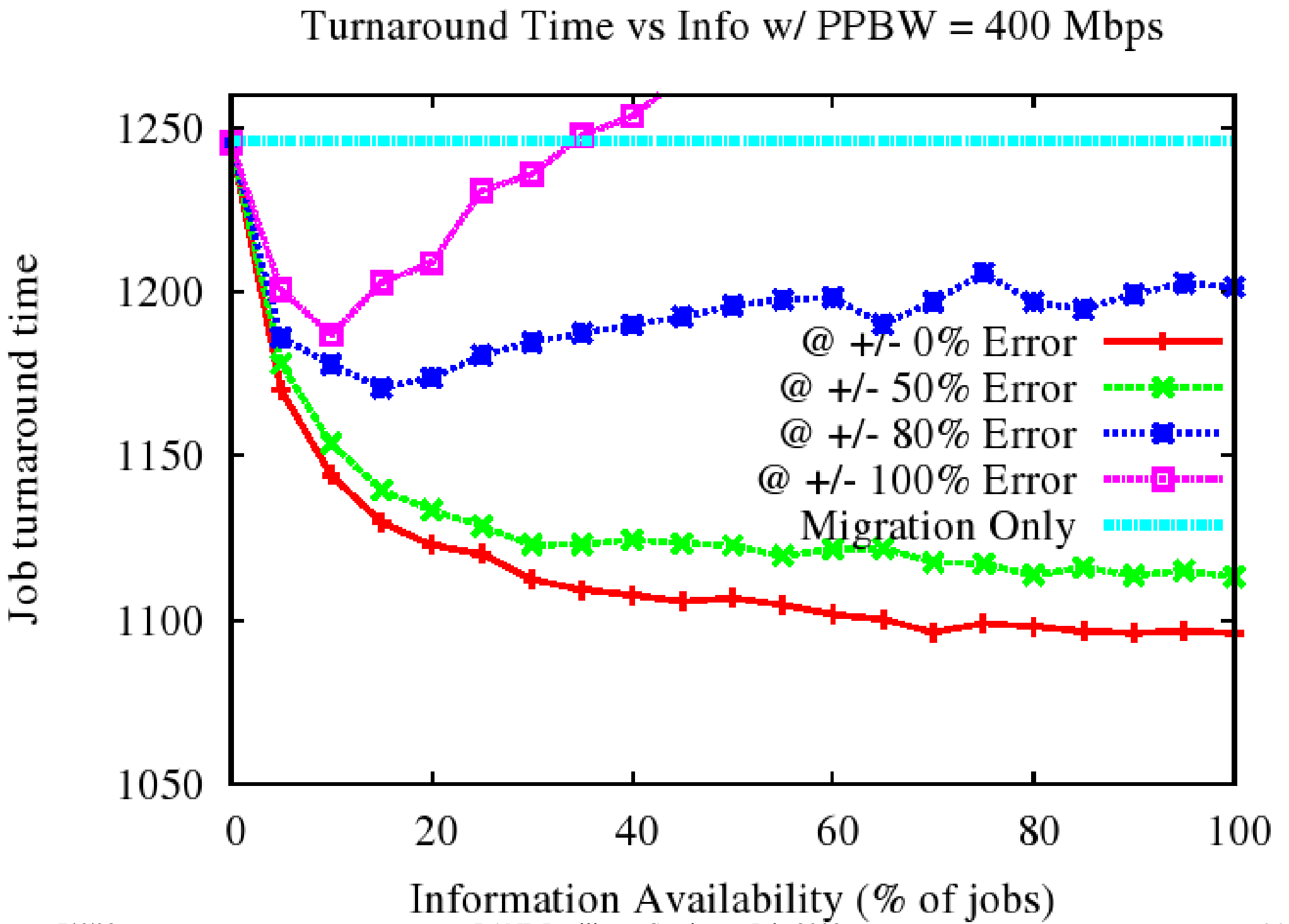

Turnaround Time vs Info vs Error w/ PPBW = 150 Mbps
TAT
Job Turnaround Time
1350
1300
1250
1200
1150
1100
1050
1000
950
1250
1200
1150
1100
1050
1000
950
100
80
60
40
20
0
+/- % Error in User Estimate
0
20
40
60
80
100
Information
Availability
(% of jobs)

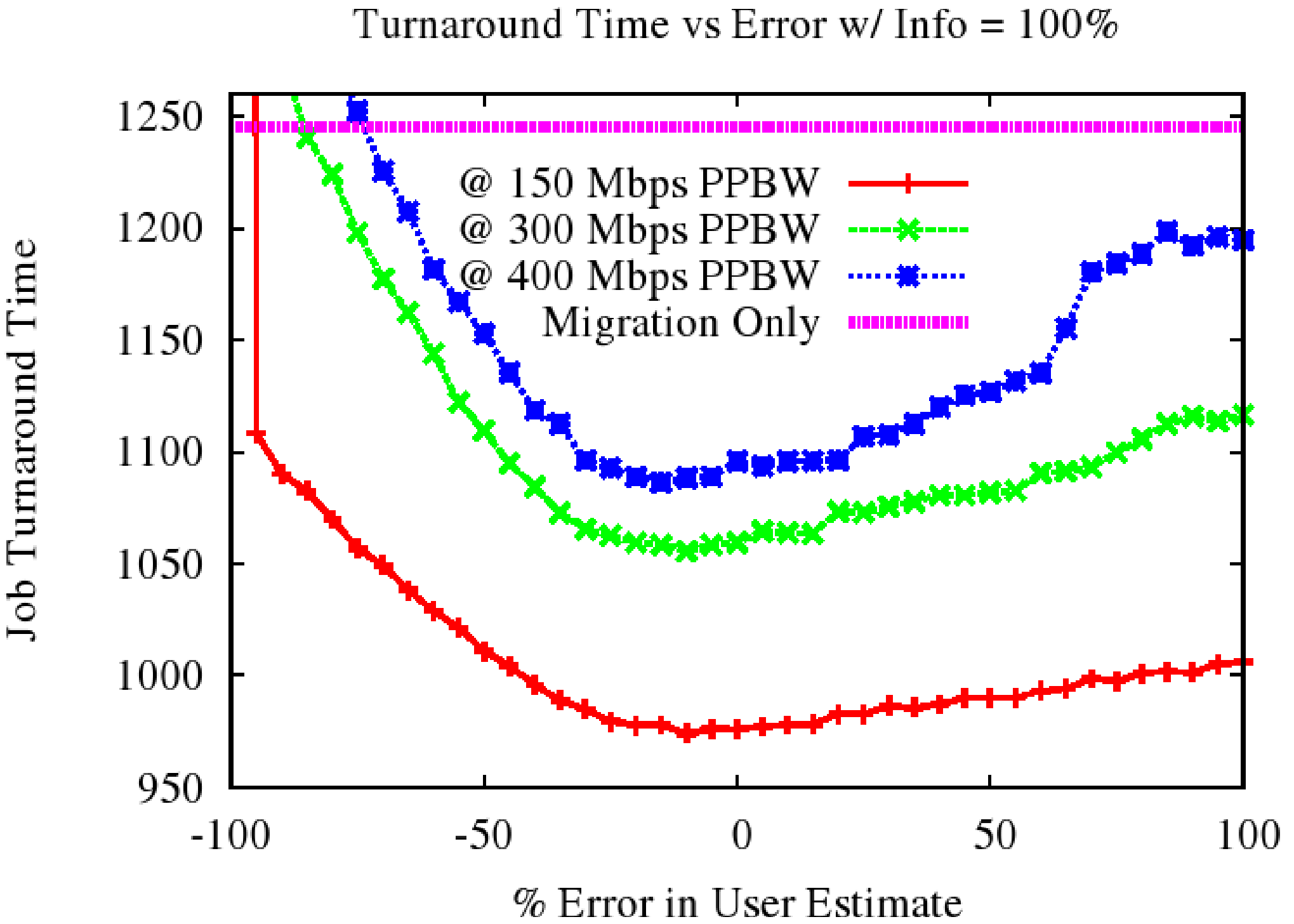

Turnaround Time vs Info w/ PPBW = 150 Mbps
Job turnaround time
1250
1200
1150
1100
1050
1000
950
0
20
40
60
80
100
Information Availability (% of jobs)
@ +/- 0% Error
@ +/- 50% Error
@ +/- 80% Error
@ +/- 100% Error
Migration Only


Turnaround Time vs Info vs Error w/ PPBW = 300
TAT
Job Turnaround Time
1350
1300
1250
1200
1150
1100
1050
1250
1200
1150
1100
1050
100
80
60
40
20
0
+/- % Error In User Estimate
0
20
40
60
80
100
Information
Availability
(% of jobs)

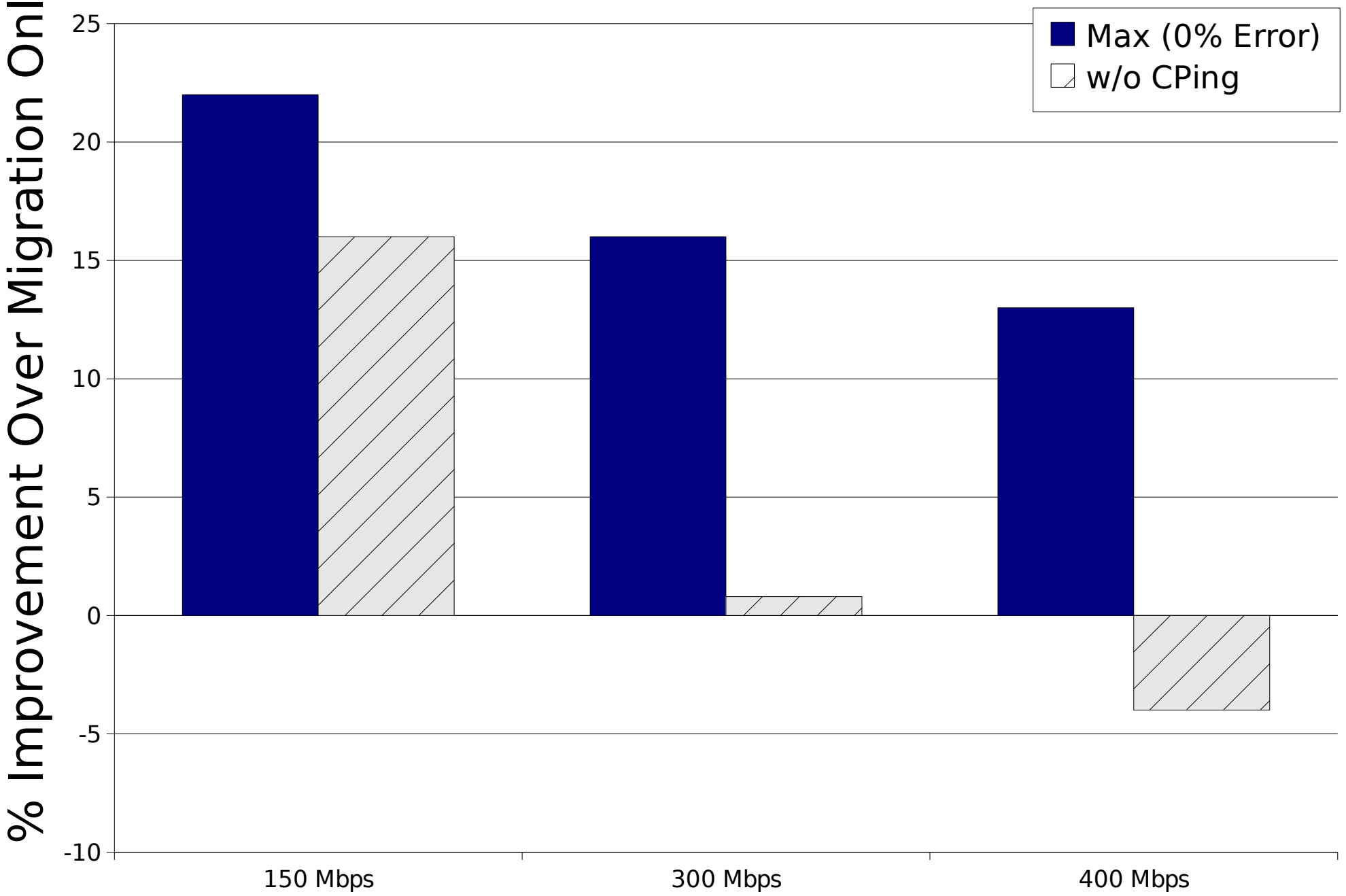

Turnaround Time vs Info w/ PPBW = 300 Mbps
Job turnaround time
1250
1200
1150
1100
1050
0
20
40
60
80
100
Information Availability (% of jobs)
@ +/- 0% Error
@ +/- 50% Error
@ +/- 80% Error
@ +/- 100% Error
Migration Only


Turnaround Time vs Info vs Error w/ PPBW = 400
TAT
Job Turnaround Time
1350
1300
1250
1200
1150
1100
1050
100
80
60
40
20
0
+/- % Error in User Estimate
0
20
40
60
80
100
Information Availability (% of jobs)
1250
1200
1150
1100
1050


Turnaround Time vs Info w/ PPBW = 400 Mbps
Job turnaround time
1250
1200
1150
1100
1050
0
20
40
60
80
100
Information Availability (% of jobs)
@ +/- 0% Error
@ +/- 50% Error
@ +/- 80% Error
@ +/- 100% Error
Migration Only

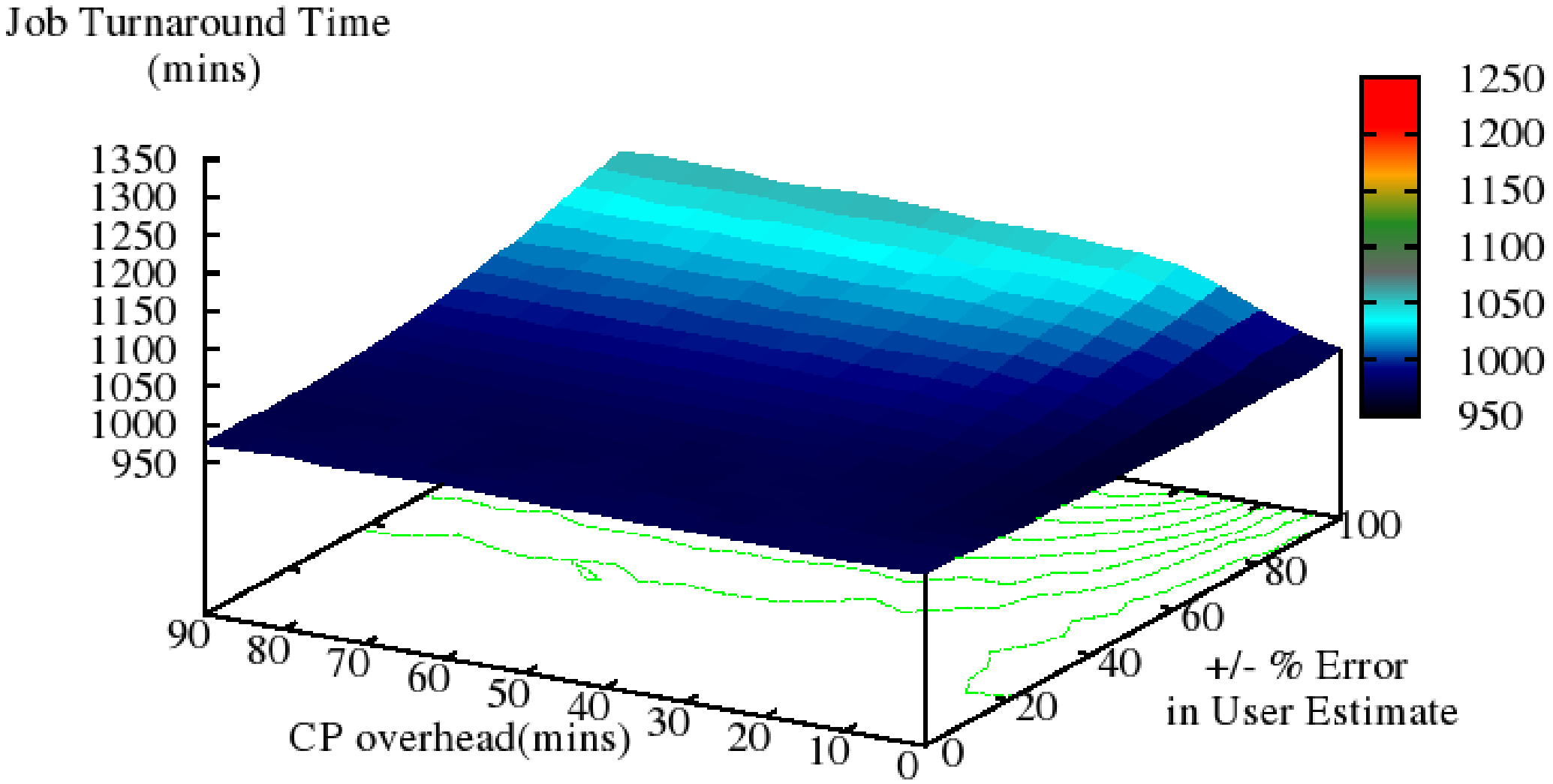

Turnaround Time vs Error w/ Info = 100%
Job Turnaround Time
1250
1200
1150
1100
1050
1000
950
-100
-50
0
50
100
% Error in User Estimate
@ 150 Mbps PPBW
@ 300 Mbps PPBW
@ 400 Mbps PPBW
Migration Only

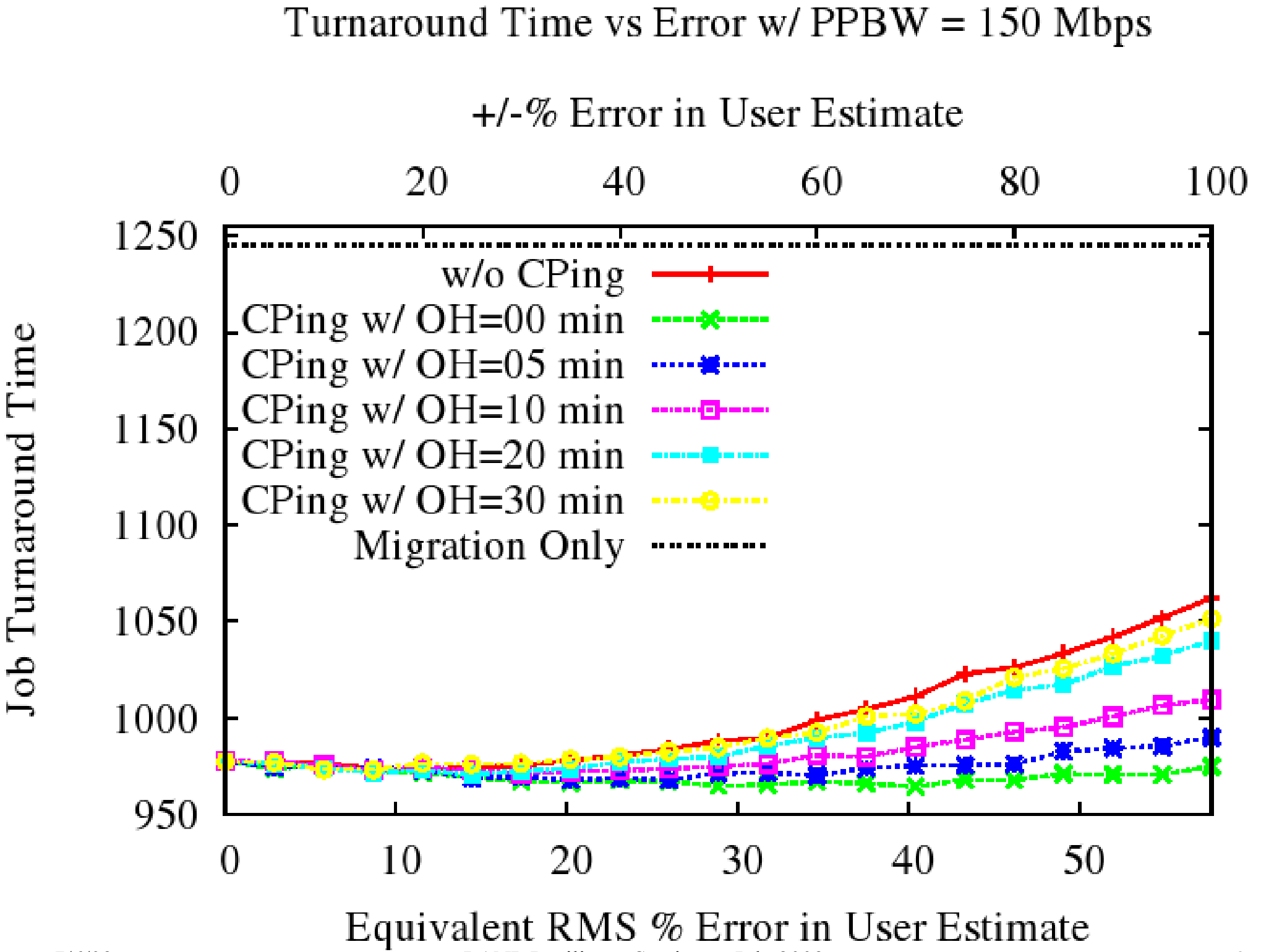
Impact of +/- 100% Estimate Inaccuracy



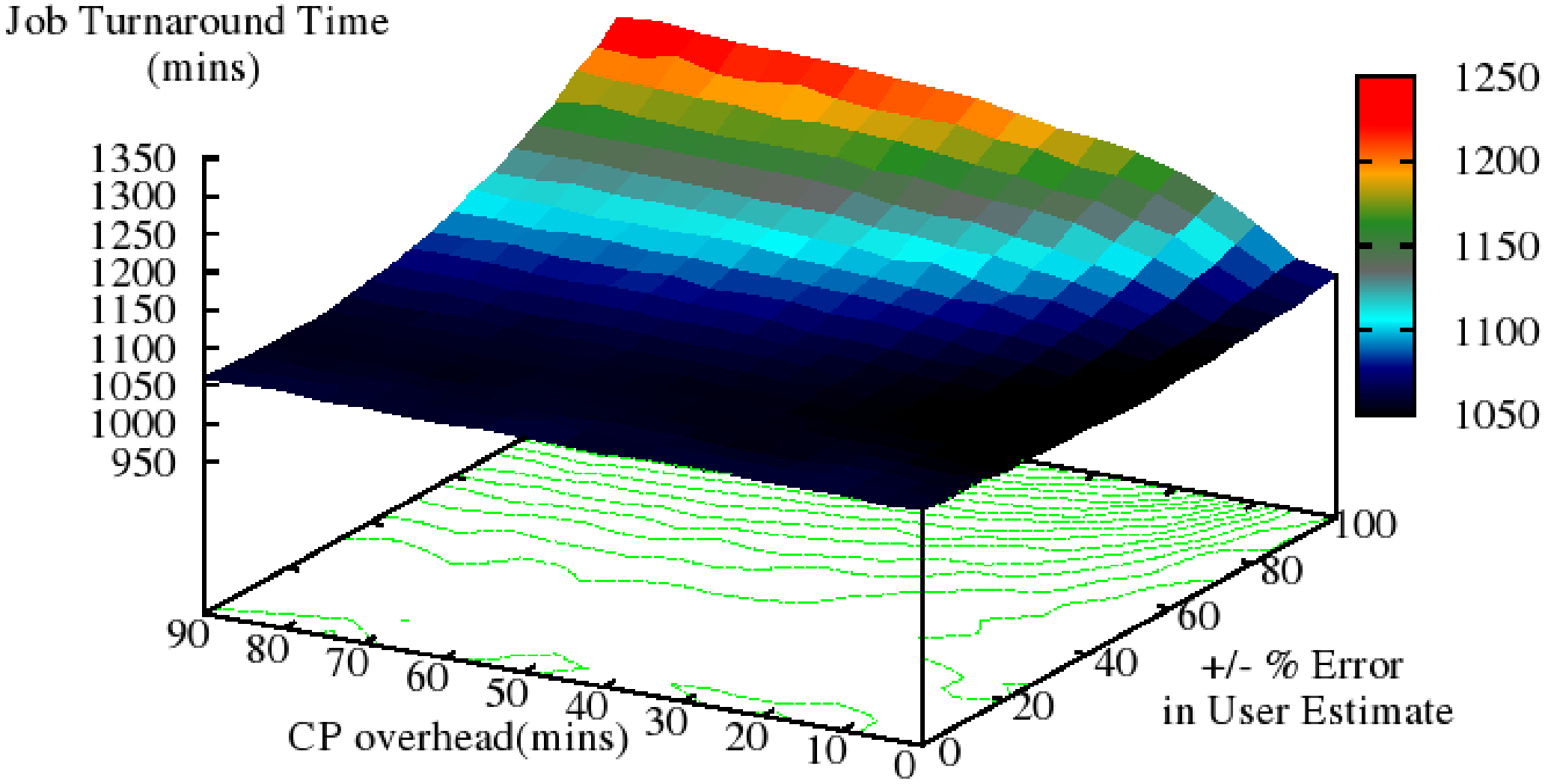
What potential gain might there be to employ **checkpointing** and **run-time job migration** to mitigate network over-subscription?

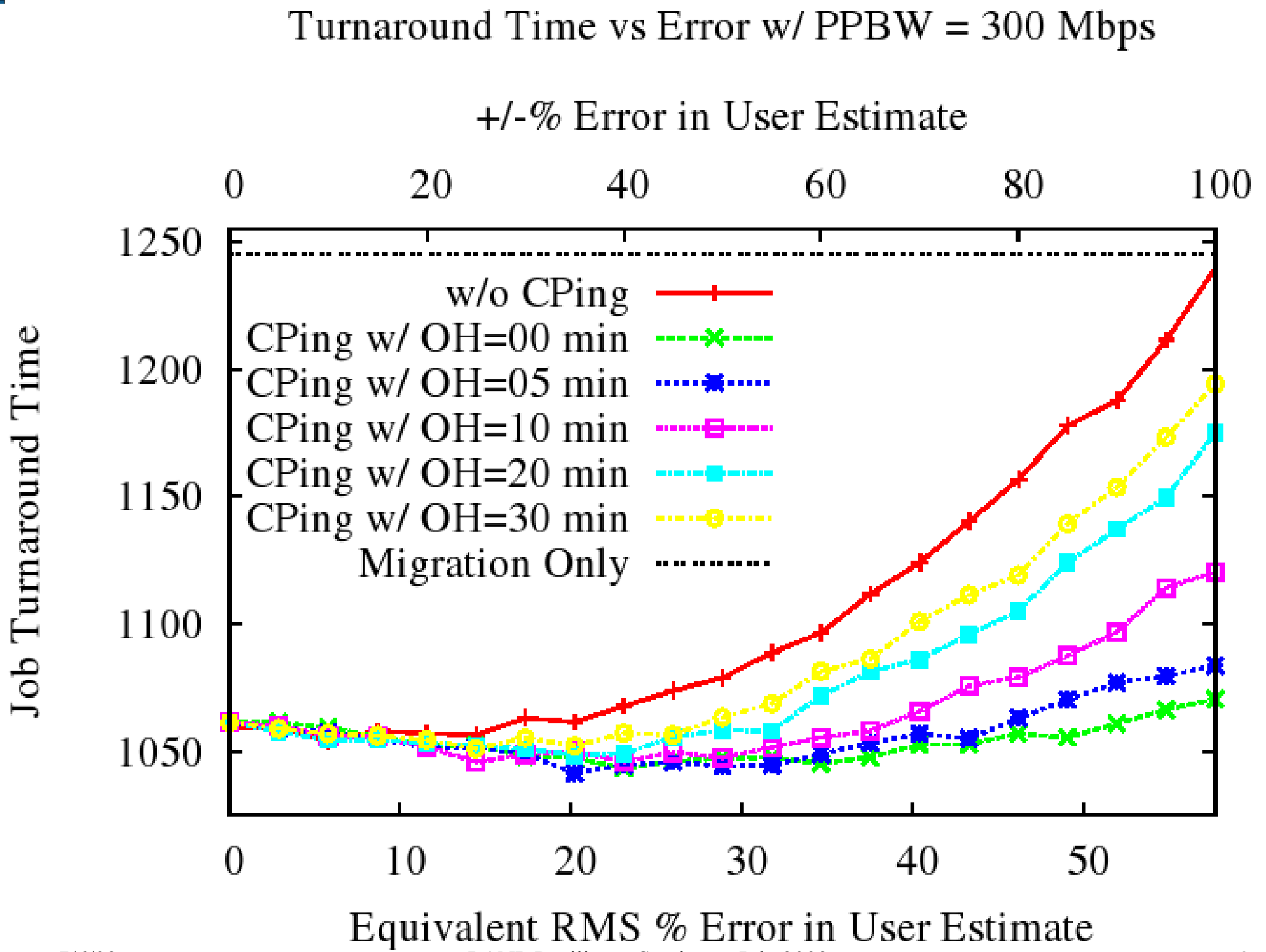
Turnaround Time vs CP Overhead vs Error w/ PPBW = 150 Mbps

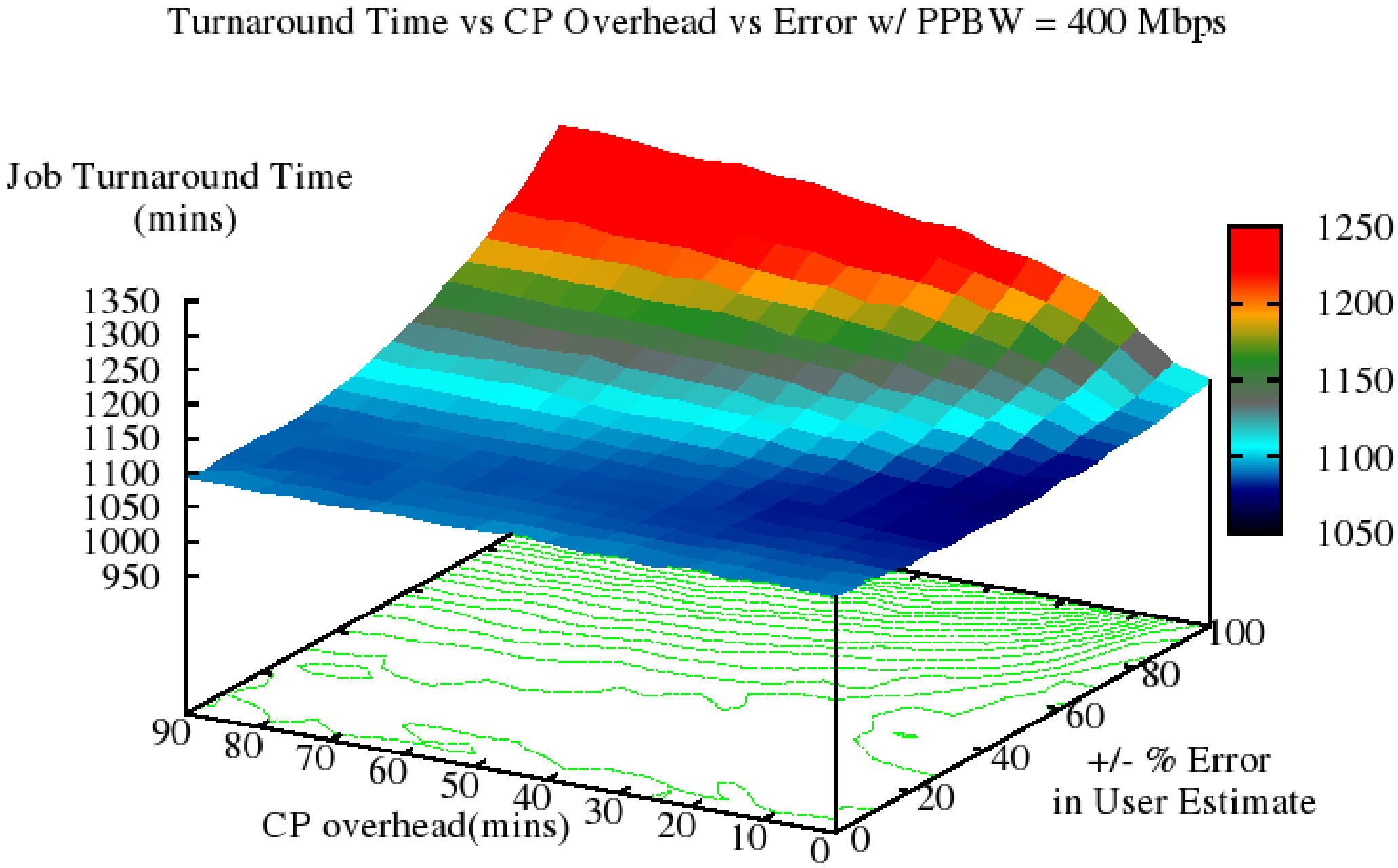


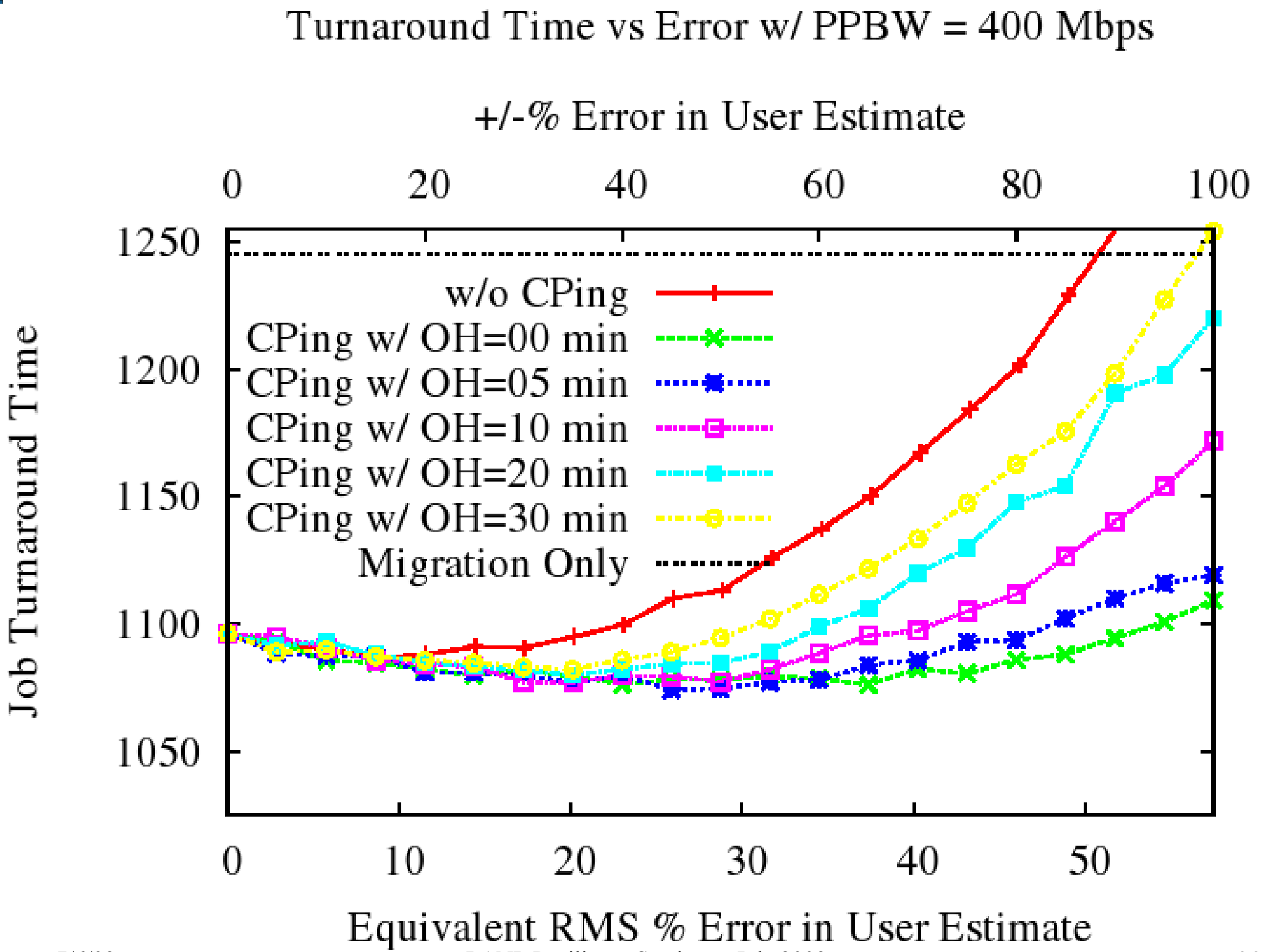

Turnaround Time vs Error w/ PPBW = 150 Mbps
+/-% Error in User Estimate
0
20
40
60
80
100
1250
1200
1150
1100
1050
1000
950
Job Turnaround Time
w/o CPing
CPing w/ OH=00 min
CPing w/ OH=05 min
CPing w/ OH=10 min
CPing w/ OH=20 min
CPing w/ OH=30 min
Migration Only
0
10
20
30
40
50
Equivalent RMS % Error in User Estimate

Turnaround Time vs CP Overhead vs Error w/ PPBW = 300 Mbps

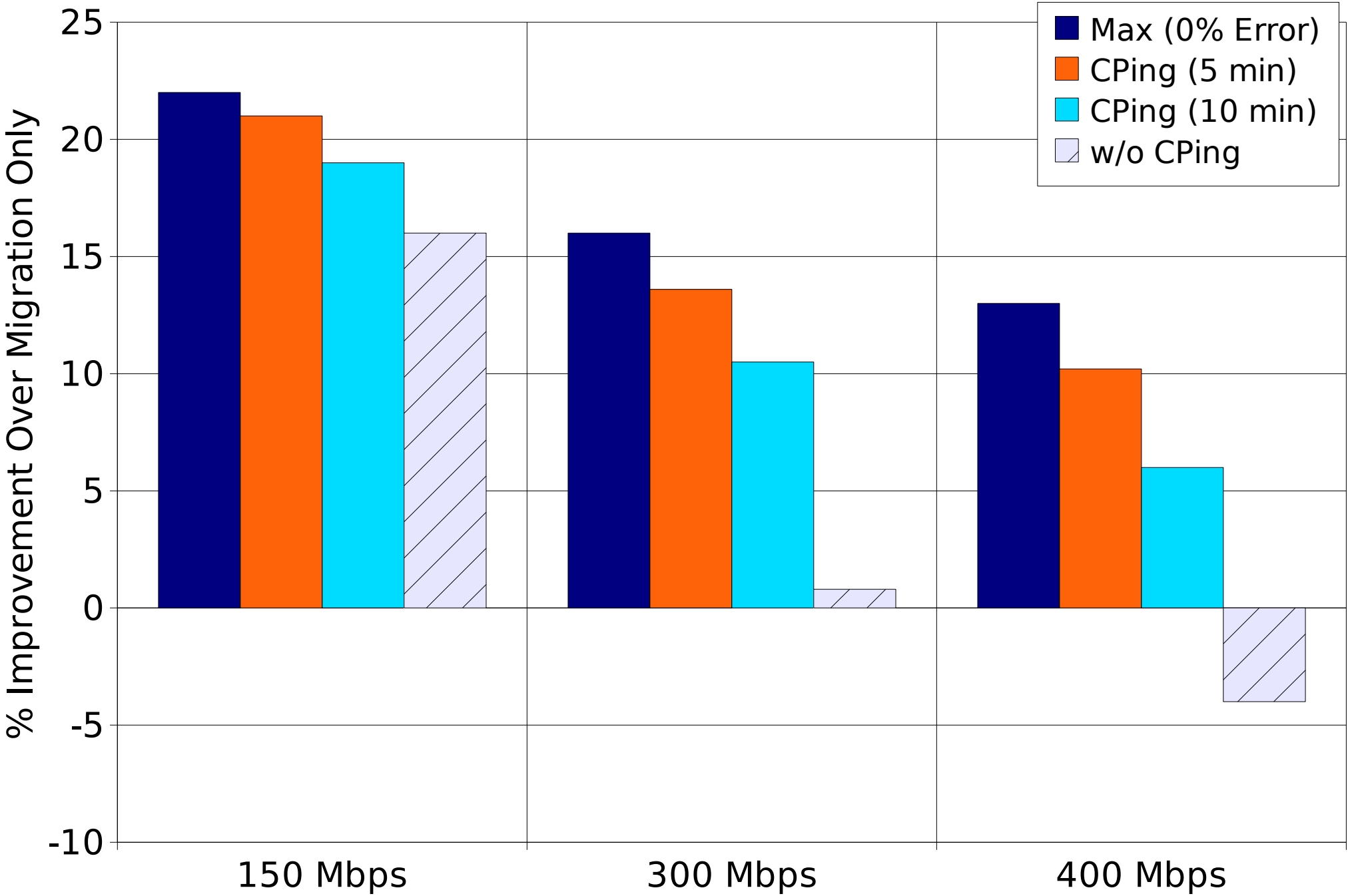


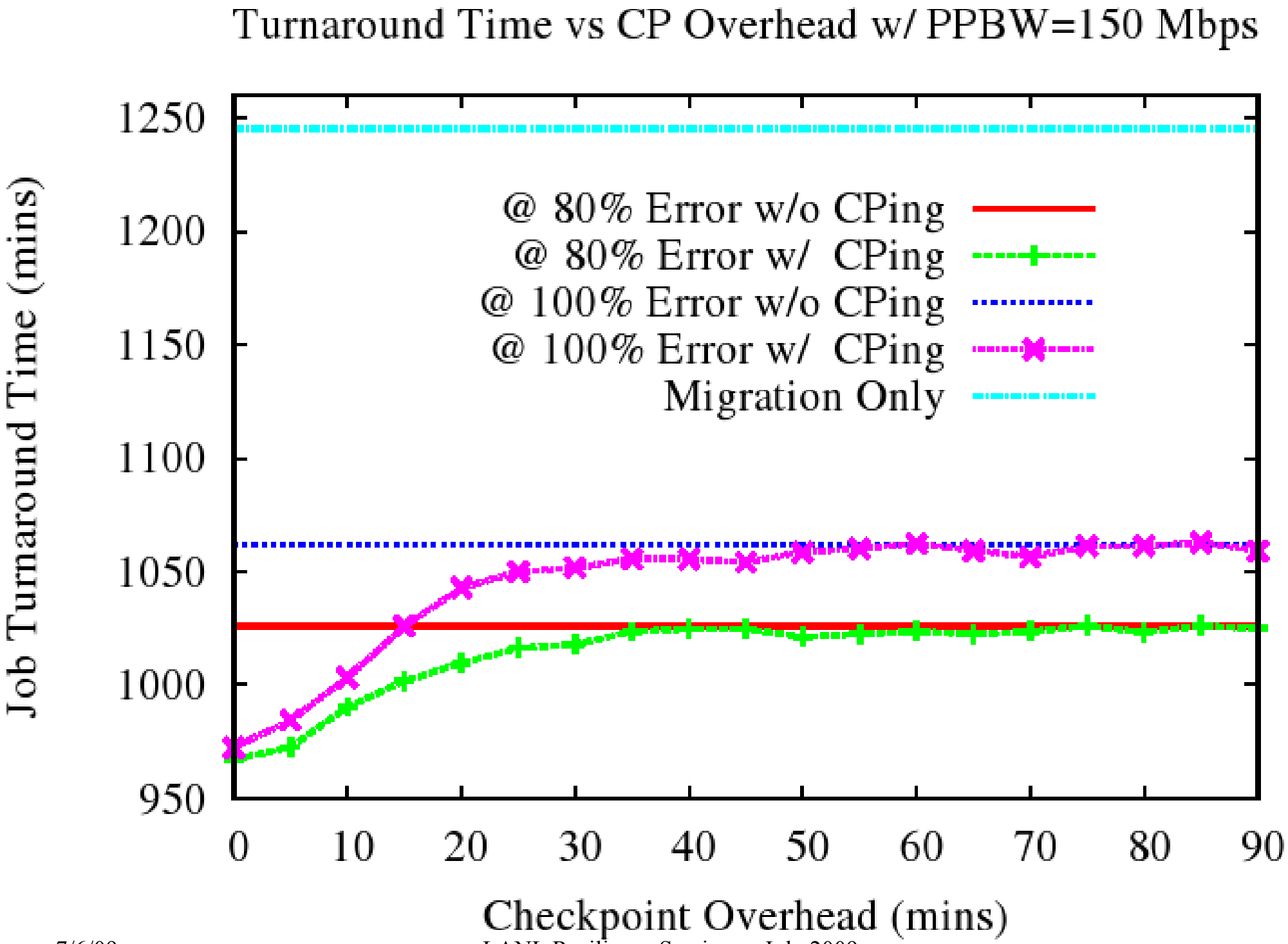

Turnaround Time vs Error w/ PPBW = 300 Mbps
+/-% Error in User Estimate
0
20
40
60
80
100
1250
1200
1150
1100
1050
Job Turnaround Time
w/o CPing
CPing w/ OH=00 min
CPing w/ OH=05 min
CPing w/ OH=10 min
CPing w/ OH=20 min
CPing w/ OH=30 min
Migration Only
0
10
20
30
40
50
Equivalent RMS % Error in User Estimate

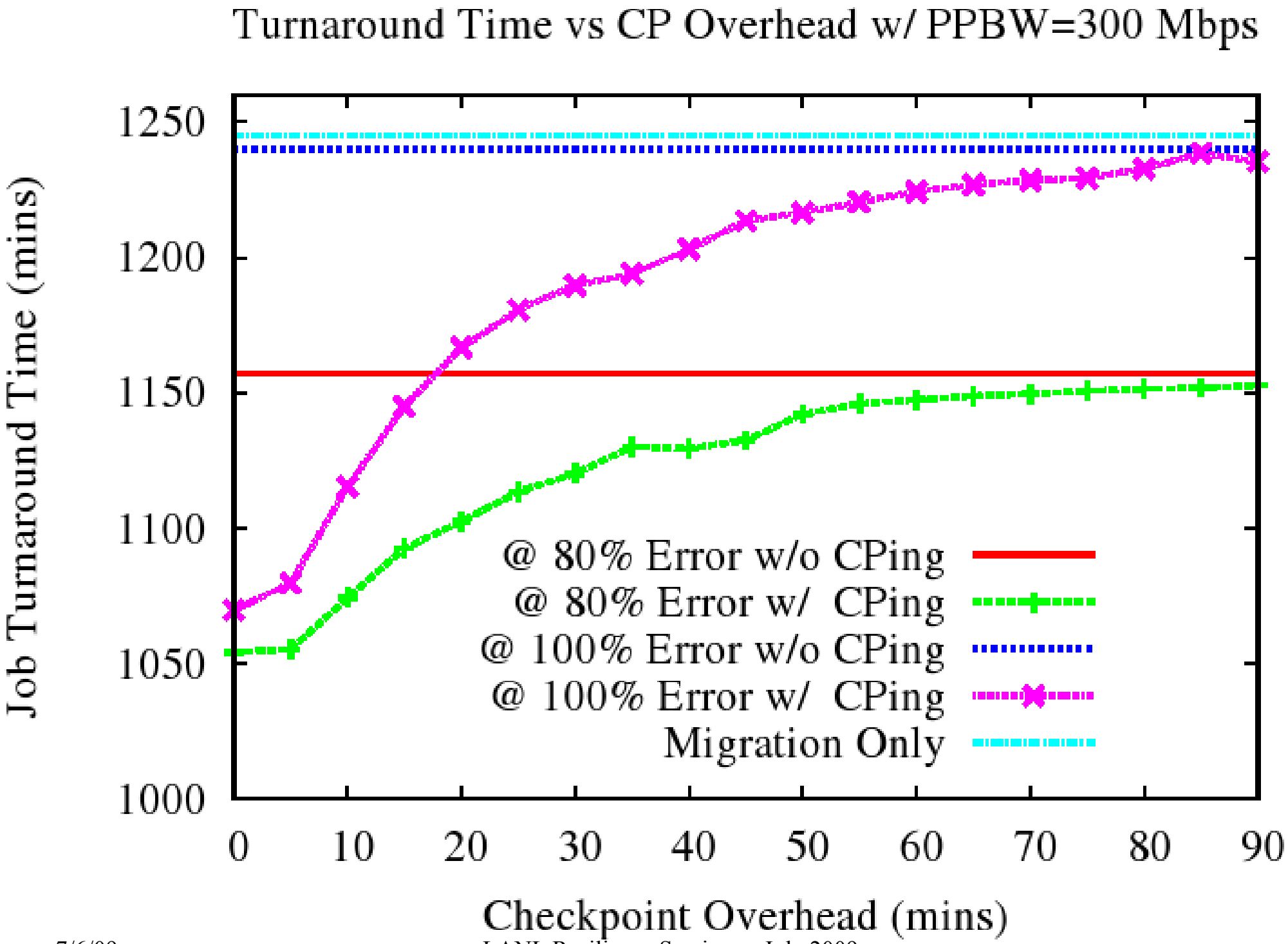

Turnaround Time vs CP Overhead vs Error w/ PPBW = 400 Mbps
Job Turnaround Time
(mins)
1350
1300
1250
1200
1150
1100
1050
1000
950
90
80
70
60
50
40
30
20
10
0
CP overhead(mins)
0
20
40
60
80
100
+/- % Error
in User Estimate
1250
1200
1150
1100
1050


Turnaround Time vs Error w/ PPBW = 400 Mbps
+/-% Error in User Estimate
0
20
40
60
80
100
1250
1200
1150
1100
1050
Job Turnaround Time
w/o CPing
CPing w/ OH=00 min
CPing w/ OH=05 min
CPing w/ OH=10 min
CPing w/ OH=20 min
CPing w/ OH=30 min
Migration Only
0
10
20
30
40
50
Equivalent RMS % Error in User Estimate

Results Summary at 5 and 10 Minute CP Overhead at 100% Estimate Error




Turnaround Time vs CP Overhead w/ PPBW=150 Mbps
Job Turnaround Time (mins)
1250
1200
1150
1100
1050
1000
950
0
10
20
30
40
50
60
70
80
90
Checkpoint Overhead (mins)
@ 80% Error w/o CPing
@ 80% Error w/ CPing
@ 100% Error w/o CPing
@ 100% Error w/ CPing
Migration Only

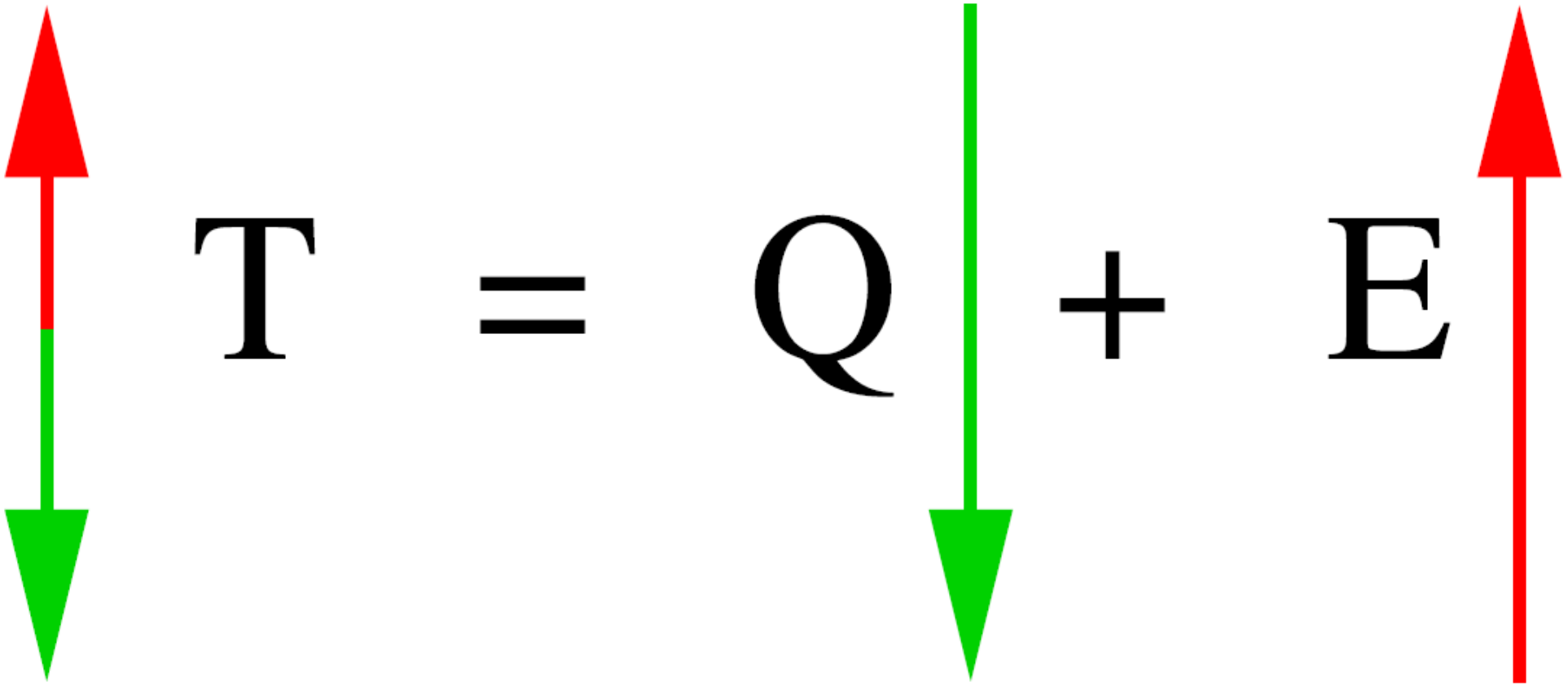

Turnaround Time vs CP Overhead w/ PPBW=300 Mbps
Job Turnaround Time (mins)
1250
1200
1150
1100
1050
1000
0
10
20
30
40
50
60
70
80
90
Checkpoint Overhead (mins)
@ 80% Error w/o CPing
@ 80% Error w/ CPing
@ 100% Error w/o CPing
@ 100% Error w/ CPing
Migration Only

Thank you!

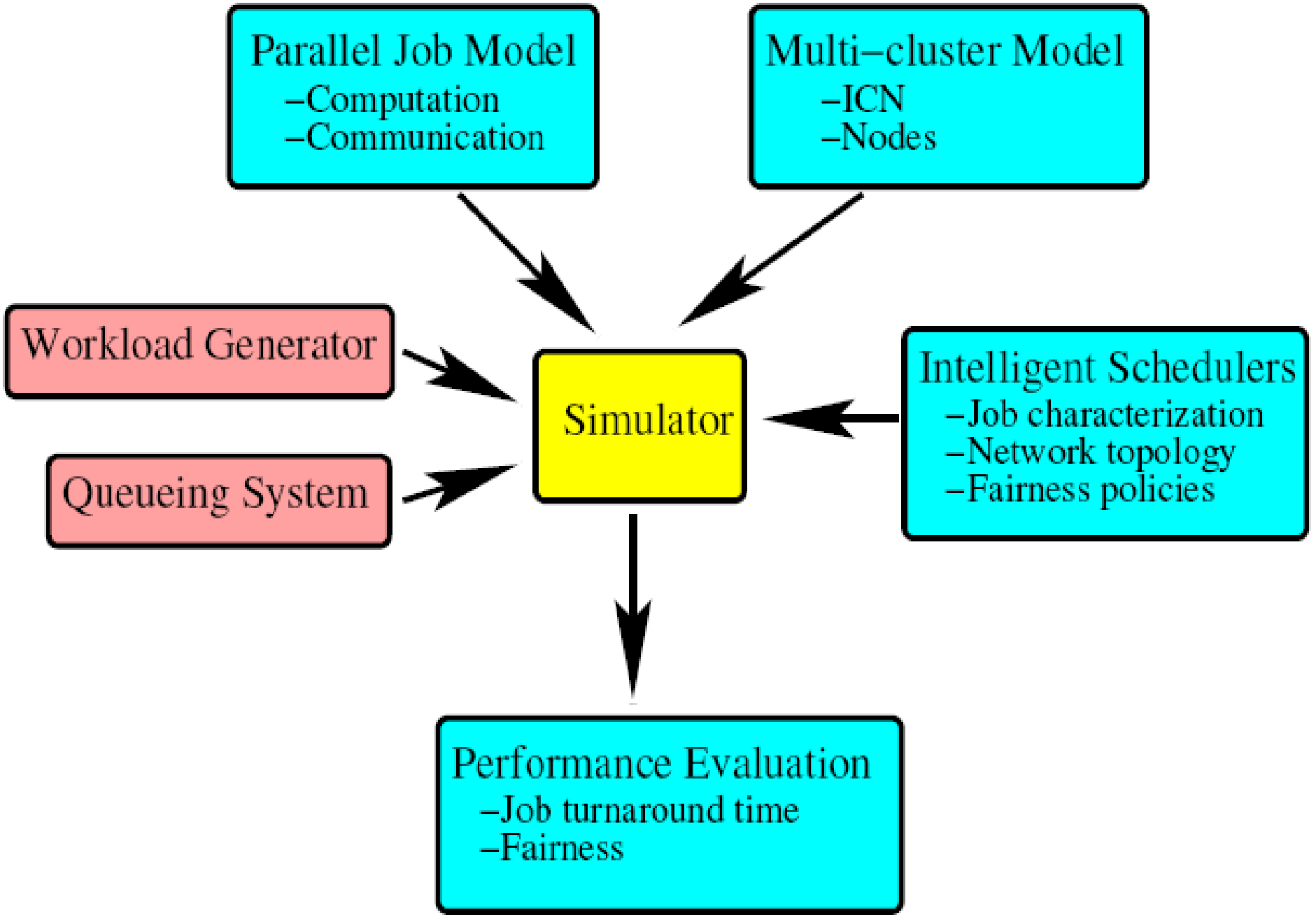
Questions?

William M. Jones
http://www.parl.clemson.edu/beosim

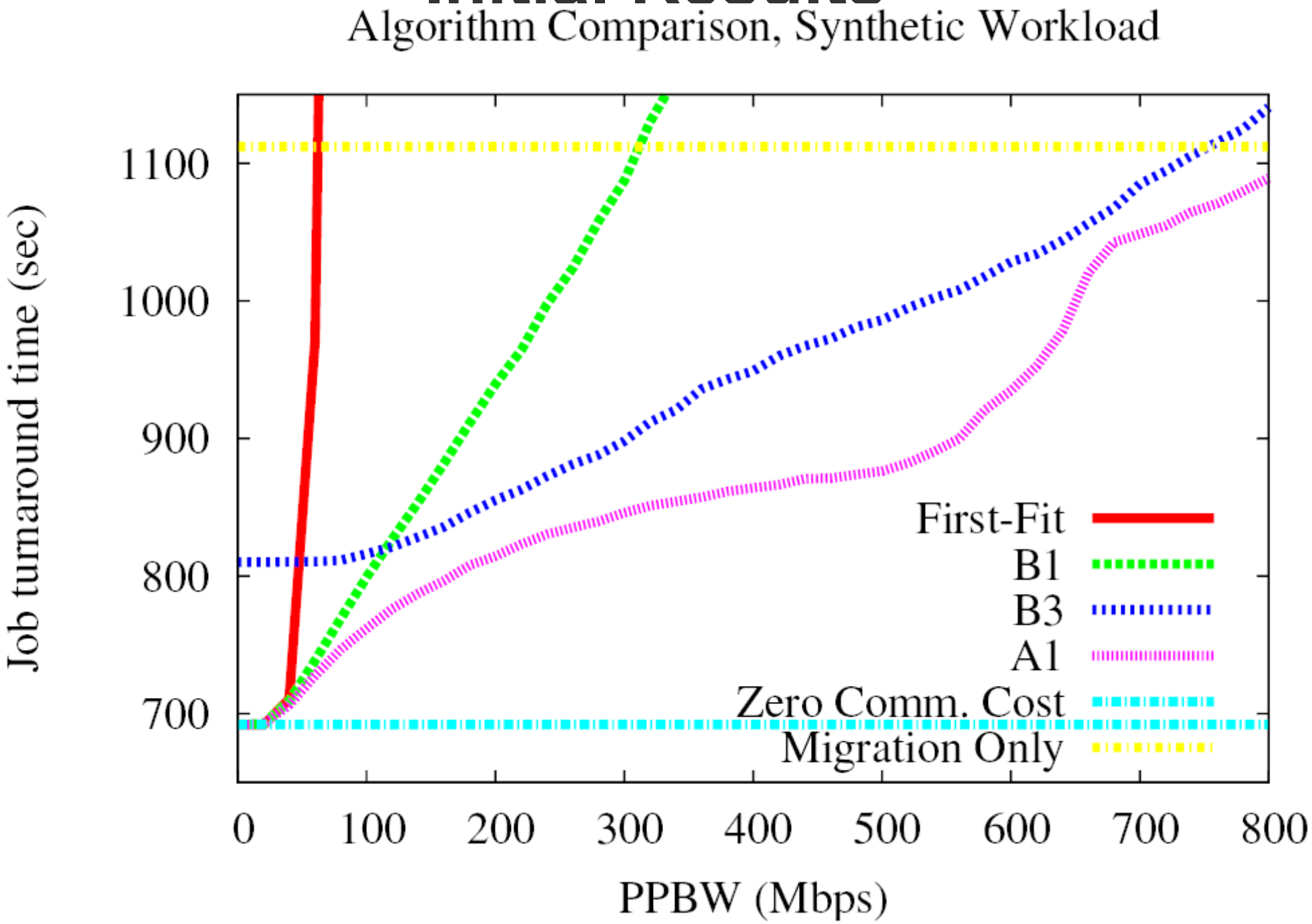
What could some error be 'good'?



Simulation Framework



Initial Results

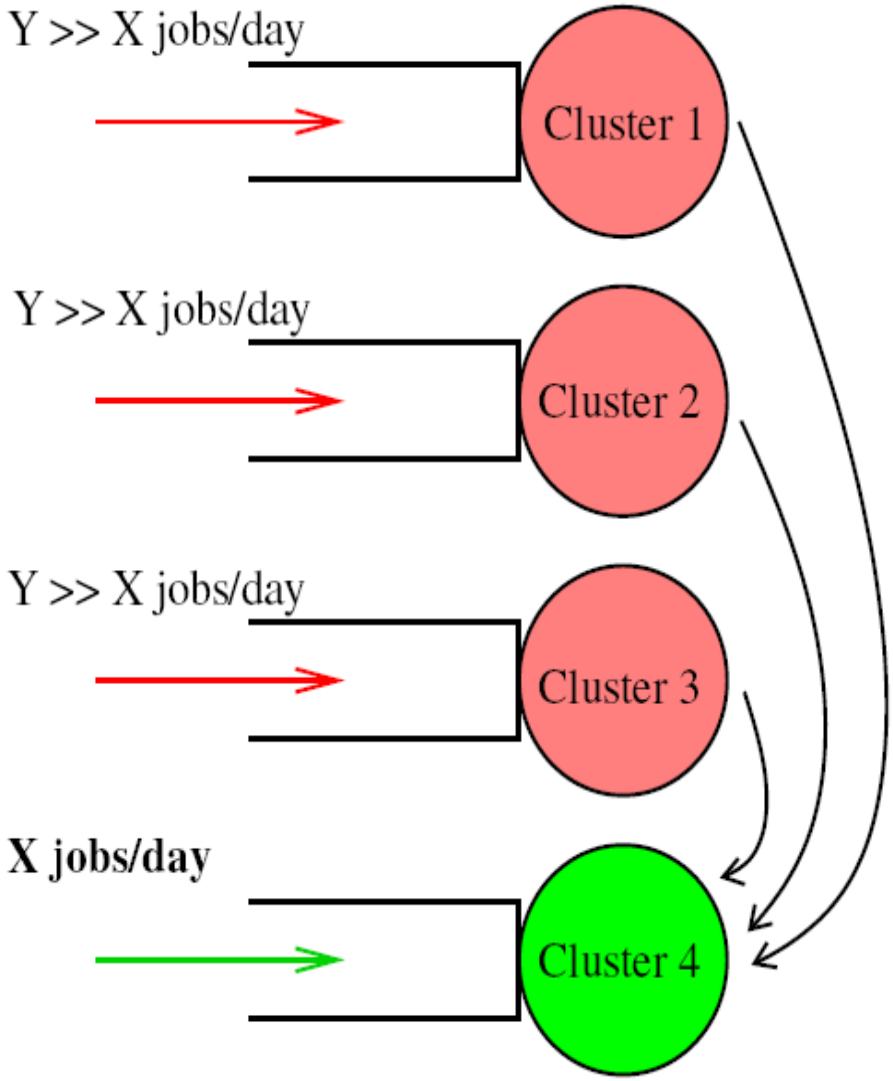


Algorithm Run-Time Analysis

Module	Complexity	Time (μSec)
A1	$O(n \cdot p^m)$	69.6
B1	$O(m \cdot log(m) + n)$	0.98
B2	$O(m \cdot log(m) + n)$	0.89
B3	$O(m \cdot log(m) + n)$	1.28
B4	$O(p)$	1.10

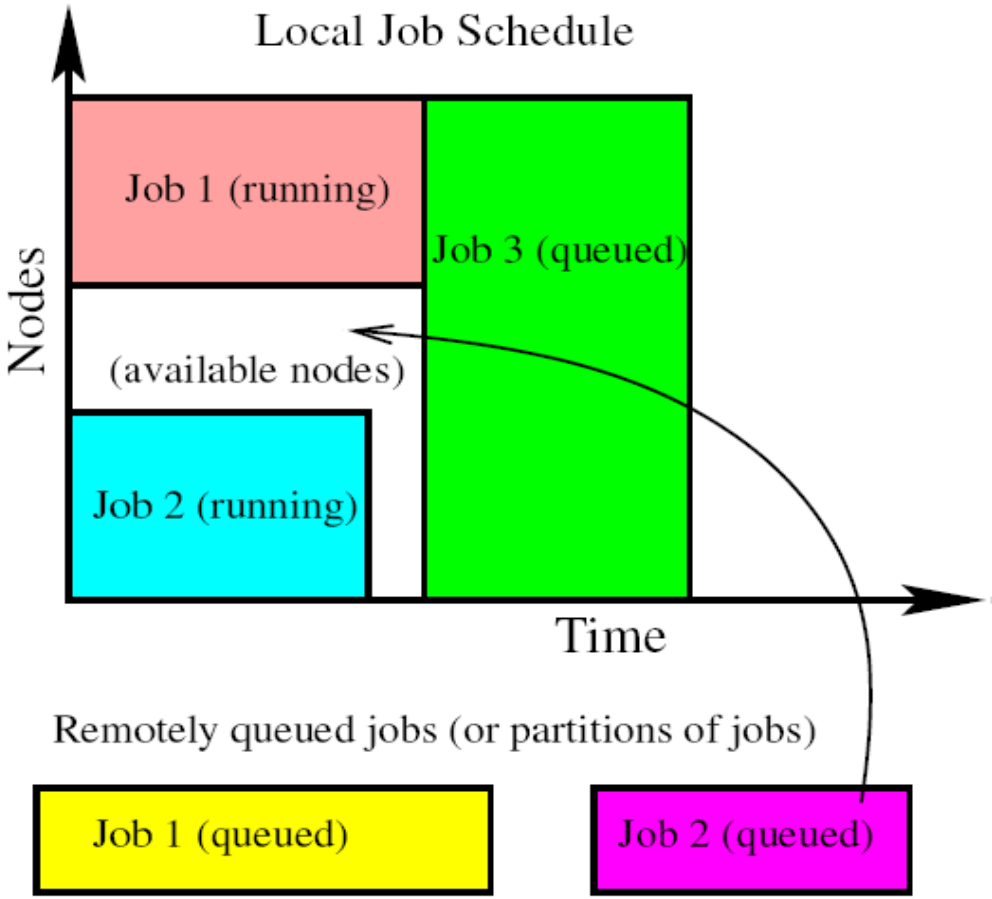
Ensuring Fairness

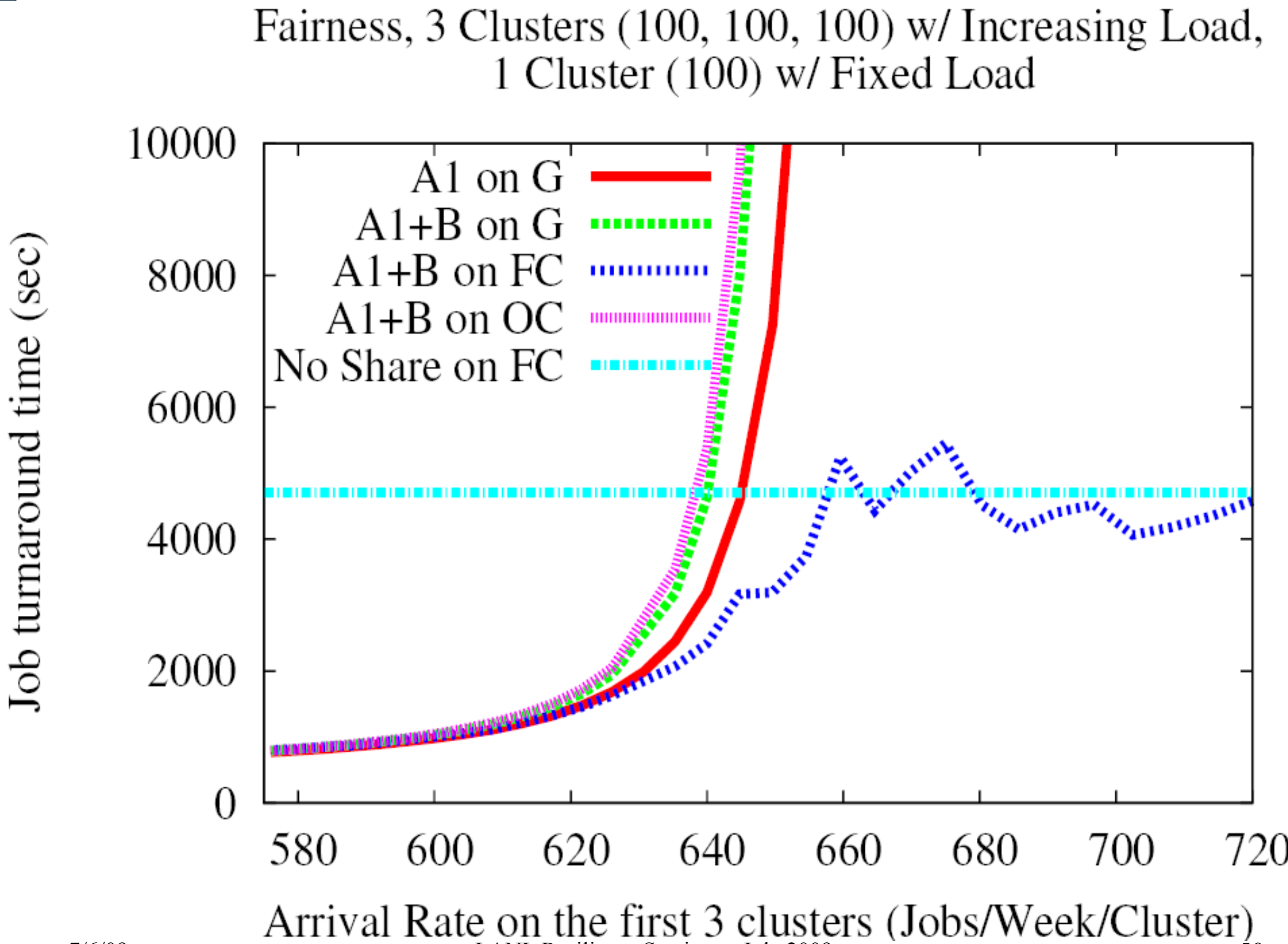
- Disparate workload intensities
- Different cluster sizes
- Unfair resource sharing
- Overload remote clusters
- Worse than not participating
- Technique to control fairness



Fairness Via Conservative Backfilling

- Out of order execution
- No delay to start time
- Two-tiered approach
- Backfill local jobs first
- Local backfill schedules
- Consider remote job backfill
- Constrain non-local node use
- Prevents local job starvation




Fairness, 3 Clusters (100, 100, 100) w/ Increasing Load,
1 Cluster (100) w/ Fixed Load
Job turnaround time (sec)
10000
8000
6000
4000
2000
0
A1 on G
A1+B on G
A1+B on FC
A1+B on OC
No Share on FC
580
600
620
640
660
680
700
720
Arrival Rate on the first 3 clusters (Jobs/Week/Cluster)

What about fault-tolerance?

Suppose you could detect that an error occurred, migrate the job, and restart the job from last checkpoint.

How quickly would you need to determine that an interrupt occurred?

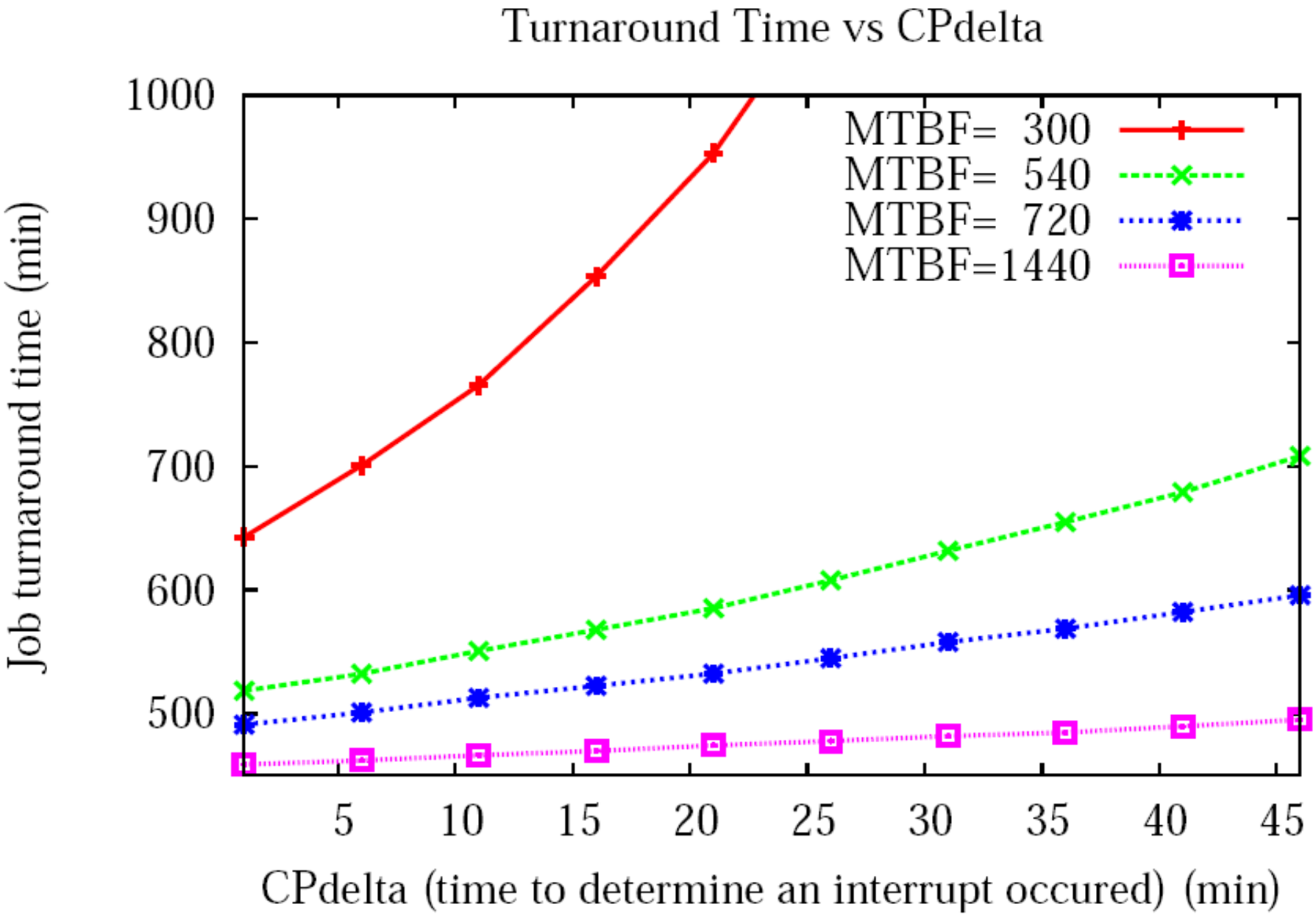


Fig. 4. Turnaround time as a function of CPdelta at various failure rates. *Note that as failures become more frequent, the overall performance becomes more sensitive to reductions in the detection latency, CPdelta.*

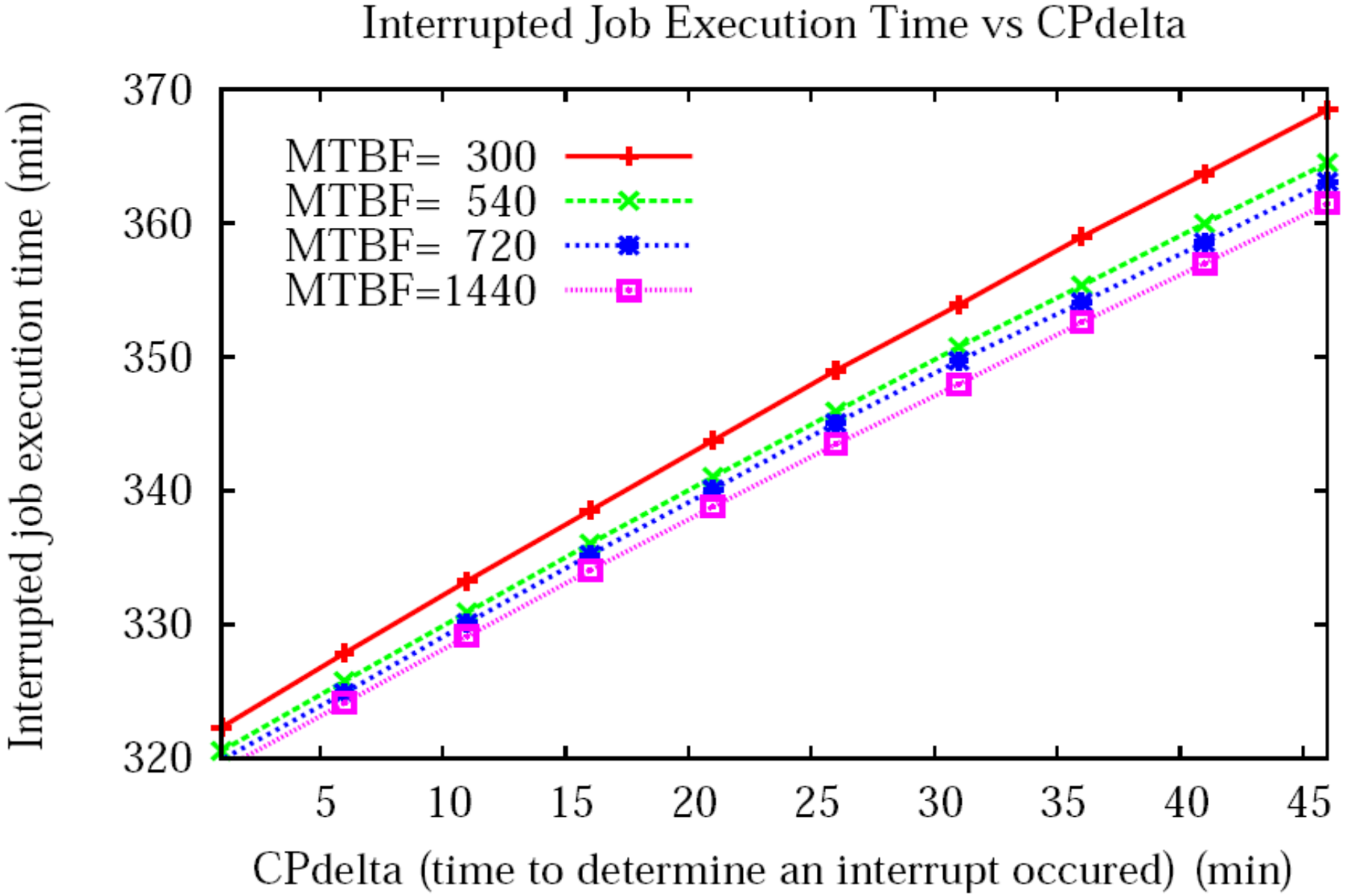


Fig. 6. Job runtime as a function of CPdelta. Here the initial job runtime distribution is the same for all classes of jobs. This is done to illustrate the expected relationship between MTBF and execution time due to multiple single-application interruptions. *Note the similar improvement due to a reduction in CPdelta.*